

Creating and Managing Tables

EX_NO:1

DATE:

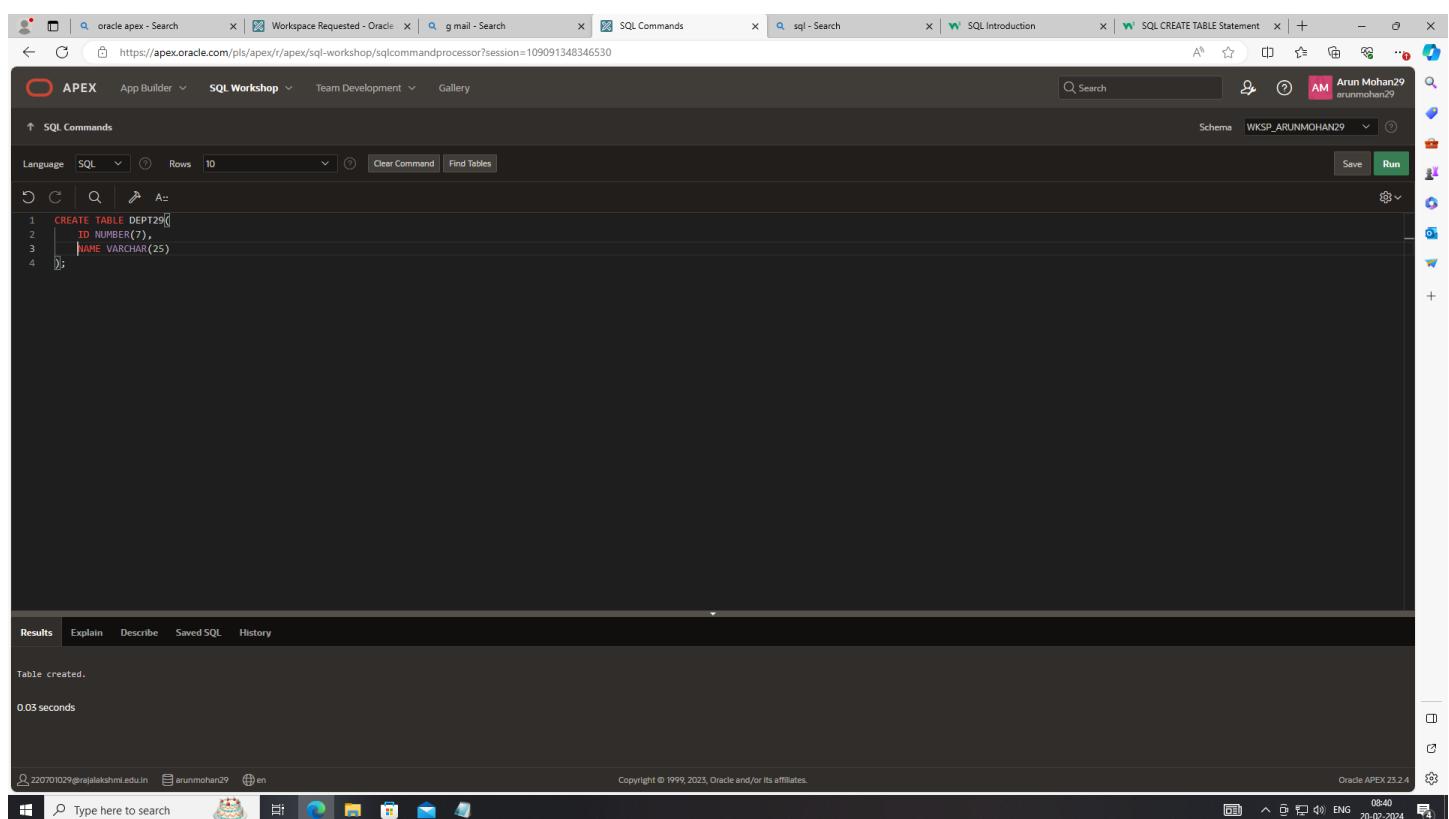
1. Create the DEPT table based on the DEPARTMENT following the table instance chart below. Confirm that the table is created.

Column name	ID	NAME
Key Type		
Nulls/Unique		
FK table		
FK column		
Data Type	Number	Varchar2
Length	7	25

QUERY:

Create table dept (id number(7),name varchar2(25));

OUTPUT:



The screenshot shows the Oracle APEX SQL Workshop interface. In the SQL Commands tab, the following SQL code is entered:

```
1 CREATE TABLE DEPT29(
2   ID NUMBER(7),
3   NAME VARCHAR(25)
4 );
```

The 'Run' button is highlighted in green at the top right of the editor. Below the editor, the Results tab is selected, showing the output:

Table created.
0.03 seconds

At the bottom, the status bar displays the user's email (220701029@rajalakshmi.edu.in) and session information (arunmohan29). The system status bar at the very bottom shows the date and time (20-02-2024 08:40).

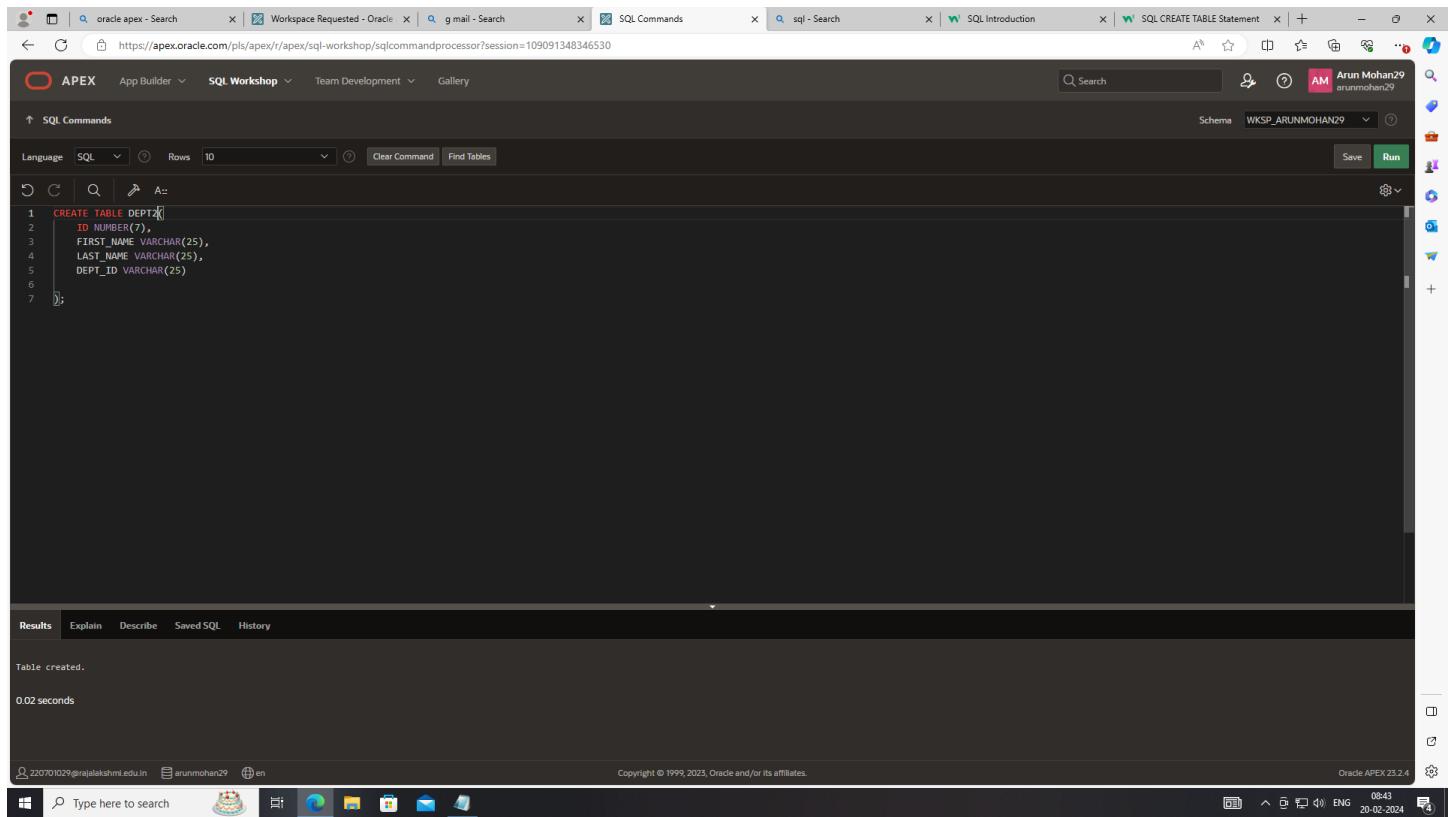
2.Create the EMP table based on the following instance chart. Confirm that the table is created.

Column name	ID	LAST_NAME	FIRST_NAME	DEPT_ID
Key Type				
Nulls/Unique				
FK table				
FK column				
Data Type	Number	Varchar2	Varchar2	Number
Length	7	25	25	7

QUERY:

```
Create table emp(id number(7),Last_Name varchar(25),First_Name varchar(25),Dept_id number(7));
```

OUTPUT:



The screenshot shows the Oracle APEX SQL Workshop interface. In the SQL Commands tab, a CREATE TABLE statement is being typed into the editor:

```
1 CREATE TABLE DEPT2
2   ID NUMBER(7),
3   FIRST_NAME VARCHAR(25),
4   LAST_NAME VARCHAR(25),
5   DEPT_ID VARCHAR(25)
6
7 );
```

After running the command, the results show:

Table created.
0.02 seconds

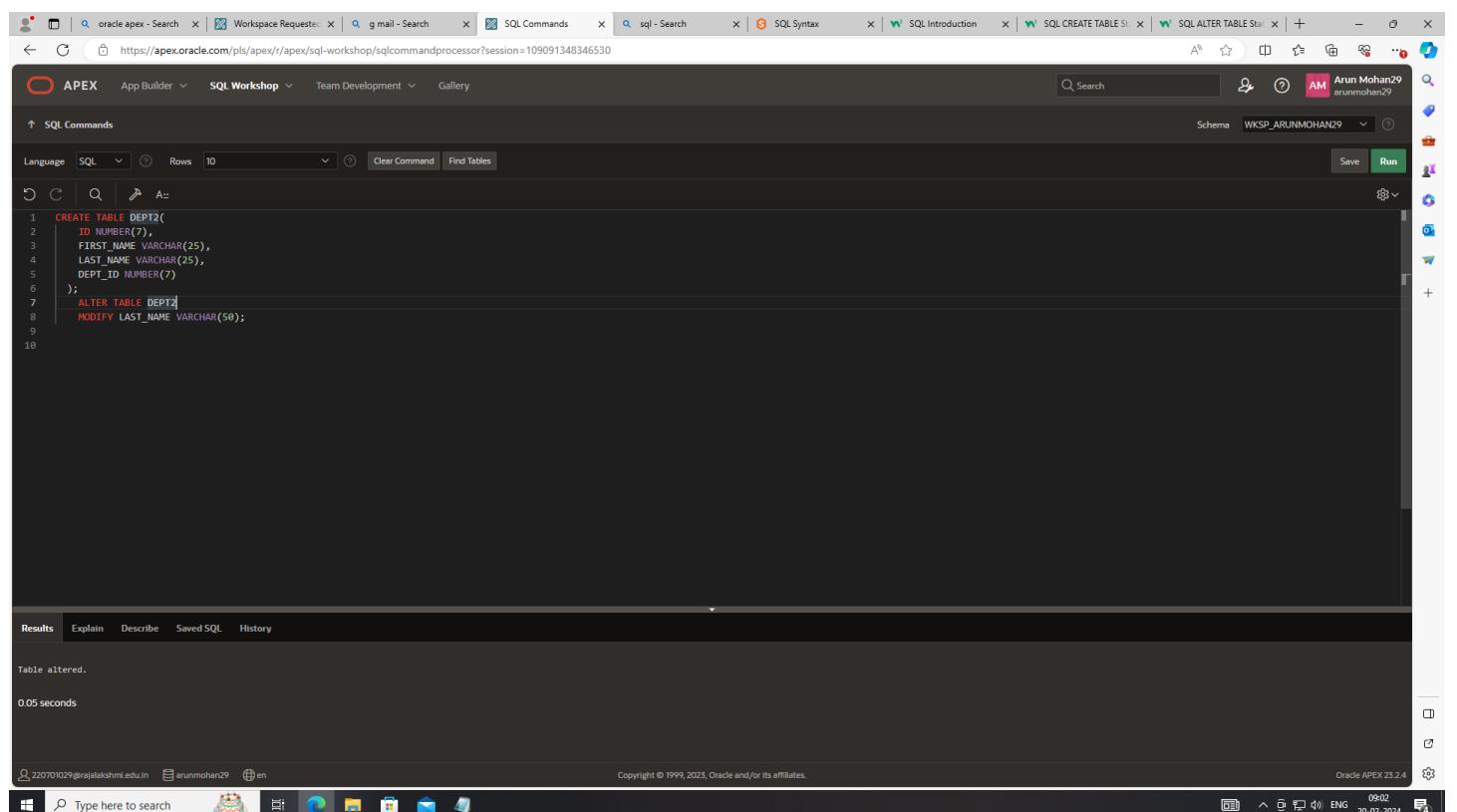
At the bottom, the status bar indicates the session is 220701029@rajalakshmi.edu.in, the schema is arunmohan29, and the date is 20-02-2024.

3.Modify the EMP table to allow for longer employee last names. Confirm the modification.(Hint: Increase the size to 50)

QUERY:

```
Alter table emp modify Last_Name varchar2(25));
```

OUTPUT:



The screenshot shows the Oracle APEX SQL Workshop interface. In the top navigation bar, there are several tabs: oracle apex - Search, Workspace Requests, g mail - Search, SQL Commands, sql - Search, SQL Syntax, SQL Introduction, SQL CREATE TABLE, and SQL ALTER TABLE. The SQL Commands tab is active. The main area displays the following SQL code:

```
1 CREATE TABLE DEPT2(
2   ID NUMBER(7),
3   FIRST_NAME VARCHAR(25),
4   LAST_NAME VARCHAR(25),
5   DEPT_ID NUMBER(7)
6 );
7 ALTER TABLE DEPT2
8   MODIFY LAST_NAME VARCHAR(50);
9
10
```

Below the code, the Results tab is selected, showing the output:

```
Table altered.
```

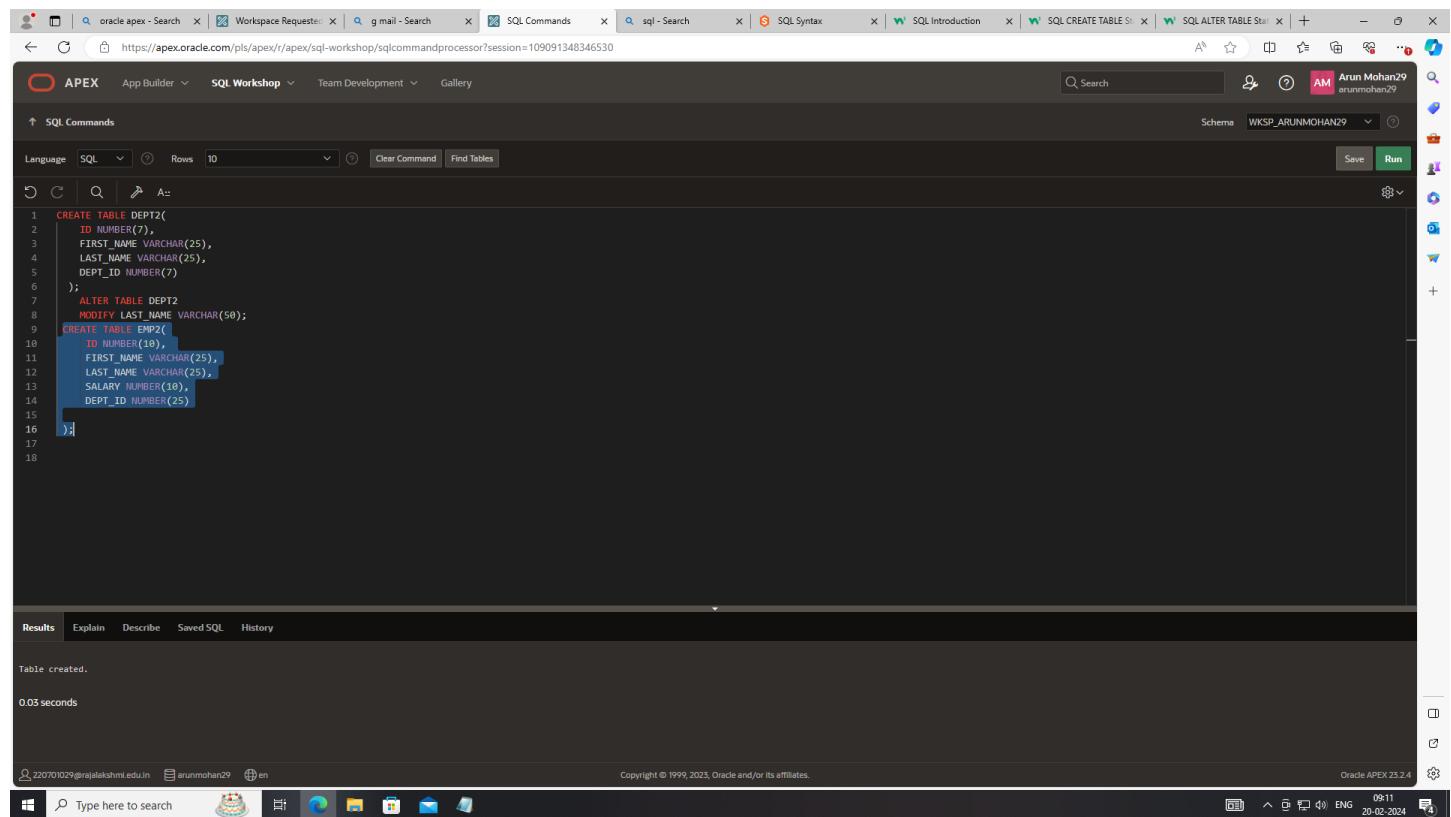
The status bar at the bottom indicates the session information: 220701029@rajalakshmi.edu.in, arunmohan29, en, Copyright © 1999, 2025, Oracle and/or its affiliates, and the system status: Oracle APEX 25.2.4.

4.Create the EMPLOYEES2 table based on the structure of EMPLOYEES table. Include Only the Employee_id, First_name, Last_name, Salary and Dept_id coloumns. Name the columns Id, First_name, Last_name, salary and Dept_id respectively.

QUERY:

```
Create table emp(id number(10),first_name varchar(25),Last_name varchar(25), Dept_id number(7));
```

OUTPUT:



The screenshot shows the Oracle APEX SQL Workshop interface. The SQL Commands tab is active, displaying the following SQL code:

```
1 CREATE TABLE DEPT2(
2   ID NUMBER(7),
3   FIRST_NAME VARCHAR(25),
4   LAST_NAME VARCHAR(25),
5   DEPT_ID NUMBER(7)
6 );
7 ALTER TABLE DEPT2
8   MODIFY LAST_NAME VARCHAR(50);
9 CREATE TABLE EMP2(
10   ID NUMBER(10),
11   FIRST_NAME VARCHAR(25),
12   LAST_NAME VARCHAR(25),
13   SALARY NUMBER(10),
14   DEPT_ID NUMBER(25)
15 );
16 );
17
18
```

The Results tab shows the output of the commands:

```
Table created.
0.03 seconds
```

At the bottom, the status bar indicates the user is 220701029@irajalakshmi.edu.in, connected to arunmohan29, and the session was run at 09:11 on 20-02-2024.

5.Drop the EMP table.

QUERY:

Drop table emp;

OUTPUT:

The screenshot shows a browser window for Oracle APEX with the URL <https://apex.oracle.com/pls/apex//apex/sql-workshop/sqlcommandprocessor?session=109091348346530>. The page title is "APEX". The main content area is titled "SQL Commands". The code entered is:

```
1 CREATE TABLE DEPT2(
2     ID NUMBER(7),
3     FIRST_NAME VARCHAR(25),
4     LAST_NAME VARCHAR(25),
5     DEPT_ID NUMBER(7)
6 );
7 ALTER TABLE DEPT2
8     MODIFY LAST_NAME VARCHAR(50);
9 CREATE TABLE EMP_2(
10    ID NUMBER(18),
11    FIRST_NAME VARCHAR(25),
12    LAST_NAME VARCHAR(25),
13    SALARY NUMBER(18),
14    DEPT_ID NUMBER(25)
15 );
16 );
17 DROP TABLE DEPT2;
```

The "Results" tab is selected at the bottom. The output shows:

```
Table dropped.
```

Execution time: 0.06 seconds

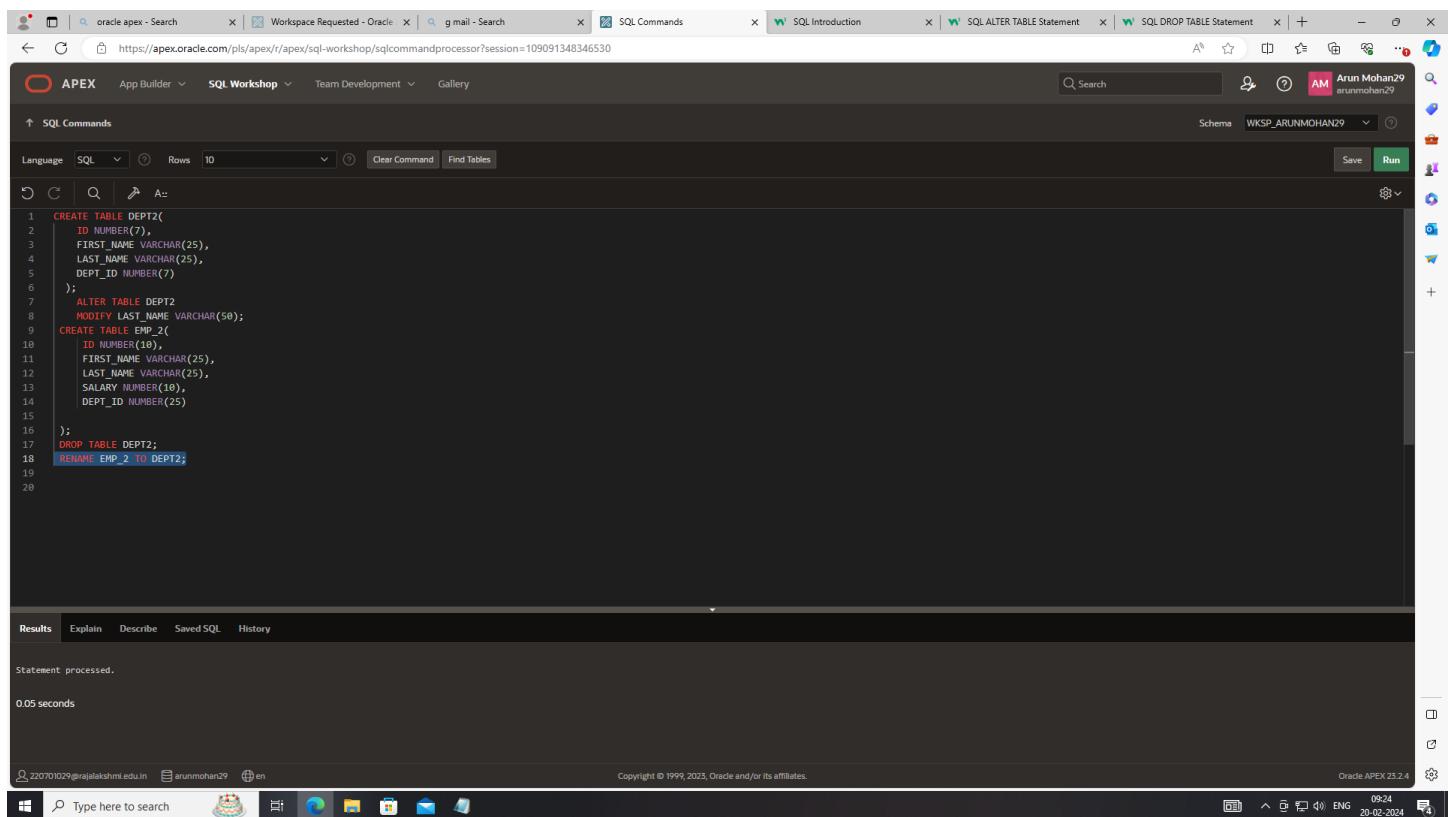
At the bottom of the browser window, the status bar displays: Copyright © 1999, 2025, Oracle and/or its affiliates. Oracle APEX 23.2.4

6.Rename the EMPLOYEES2 table as EMP.

QUERY:

Rename emp_2 to dept2;

OUTPUT:



The screenshot shows the Oracle APEX SQL Workshop interface. The SQL Commands tab is active, displaying the following SQL code:

```
1 CREATE TABLE DEPT2(
2   ID NUMBER(7),
3   FIRST_NAME VARCHAR(25),
4   LAST_NAME VARCHAR(25),
5   DEPT_ID NUMBER(7)
6 );
7 ALTER TABLE DEPT2
8   MODIFY LAST_NAME VARCHAR(50);
9 CREATE TABLE EMP_2(
10   ID NUMBER(10),
11   FIRST_NAME VARCHAR(25),
12   LAST_NAME VARCHAR(25),
13   SALARY NUMBER(10),
14   DEPT_ID NUMBER(25)
15 );
16 );
17 DROP TABLE DEPT2;
18 RENAME EMP_2 TO DEPT2;
19
20
```

The results section shows the output of the executed statements:

```
Statement processed.
0.05 seconds
```

At the bottom, the browser status bar indicates the session details and the Oracle APEX version.

7.Add a comment on DEPT and EMP tables. Confirm the modification by describing the table.

QUERY:

comment on table dept is 'Department info';
comment on table emp is Employee info';

OUTPUT:

7.Add a comment on DEPT and EMP tables. Confirm the modification by describing the table.

QUERY:

Describe Dept2;

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes links for Oracle Apex, App Builder, SQL Workshop, Team Development, and Gallery. The current tab is SQL Workshop. The right side of the screen displays the user profile of Arun Mohan29 (arunmohan29) and the schema WKS_ARUNMOHAN29. The main workspace contains the following SQL code:

```
1 CREATE TABLE DEPT2(
2     ID NUMBER(7),
3     FIRST_NAME VARCHAR(25),
4     LAST_NAME VARCHAR(25),
5     DEPT_ID NUMBER(7)
6 );
7 
8 ALTER TABLE DEPT2
9     MODIFY LAST_NAME VARCHAR(50);
10 
11 CREATE TABLE EMP_2(
12     ID NUMBER(18),
13     FIRST_NAME VARCHAR(25),
14     LAST_NAME VARCHAR(25),
15     SALARY NUMBER(18),
16     DEPT_ID NUMBER(25)
17 );
18 
19 DROP TABLE DEPT2;
20 RENAME EMP_2 TO DEPT2;
21 ALTER TABLE DEPT2
22     DROP COLUMN FIRST_NAME;
23 
24 | DESCRIBE DEPT2;
```

Below the code, there is a results grid showing the structure of the DEPT2 table:

Name	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comments
DEPT2	ID	NUMBER	-	10	0	-	✓	-	-
	LAST_NAME	VARCHAR2	25	-	-	-	✓	-	-
	SALARY	NUMBER	-	10	0	-	✓	-	-
	DEPT_ID	NUMBER	-	25	0	-	✓	-	-

The status bar at the bottom indicates the session ID (220701029@ajalakshmi.edu.in), the schema (arunmohan29), and the date (20-02-2024). The bottom right corner shows the Oracle APEX version (23.2.4).

8.Drop the First_name column from the EMP table and confirm it.

QUERY:

```
Alter table emp drop column first_name;
```

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. The SQL Commands tab is active, displaying the following SQL code:

```
1 CREATE TABLE DEPT2(
2   ID NUMBER(7),
3   FIRST_NAME VARCHAR(25),
4   LAST_NAME VARCHAR(25),
5   DEPT_ID NUMBER(7)
6 );
7 ALTER TABLE DEPT2
8   MODIFY LAST_NAME VARCHAR(50);
9 CREATE TABLE EMP_2(
10   ID NUMBER(10),
11   FIRST_NAME VARCHAR(25),
12   LAST_NAME VARCHAR(25),
13   SALARY NUMBER(10),
14   DEPT_ID NUMBER(25)
15 );
16 DROP TABLE DEPT2;
17 RENAME EMP_2 TO DEPT2;
18 ALTER TABLE DEPT2
19   DROP COLUMN FIRST_NAME;
20
21
22
```

The command at line 19, `DROP COLUMN FIRST_NAME;`, is highlighted in blue. The results section below shows the output:

```
Table altered.  
0.08 seconds
```

At the bottom, the status bar indicates the session details: 220701029@rajalakshmi.edu.in, arunmohan29, en, Copyright © 1999, 2023, Oracle and/or its affiliates, Oracle APEX 23.2.4, and the system date/time: 09:25 20-02-2024.

Evaluation Procedure	Marks awarded
Query(5)	
Execution (5)	
Viva(5)	
Total (15)	
Faculty Signature	

RESULT:

MANIPULATING DATA

EX_NO:2

DATE:

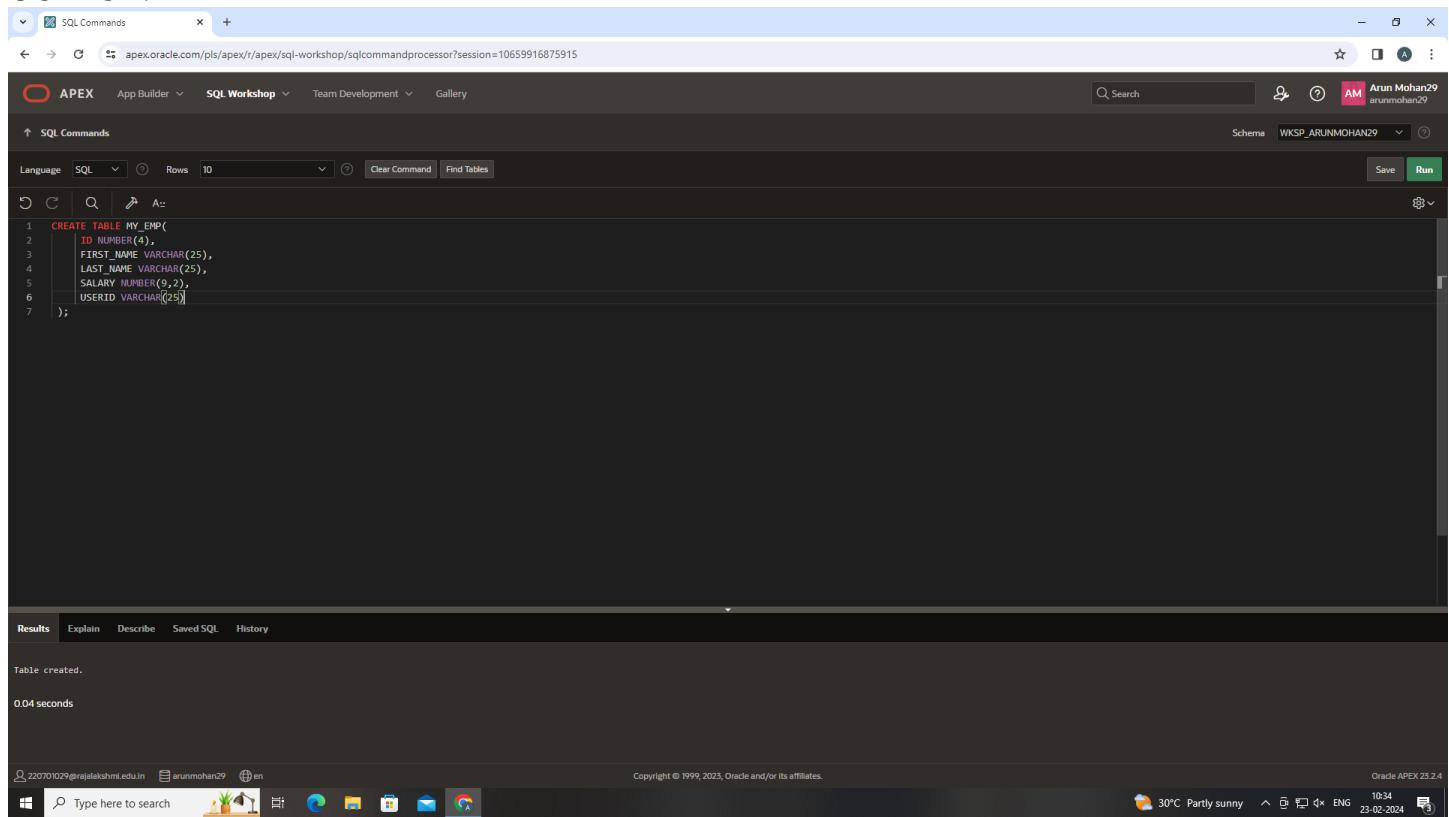
1.Create MY_EMPLOYEE table with the following structure

NAME	NULL?	TYPE
ID	Not null	Number(4)
Last_name		Varchar(25)
First_name		Varchar(25)
Userid		Varchar(25)
Salary		Number(9,2)

QUERY:

```
Create table my_employee( ID number(4),First_name varchar(25),Last_name varchar(25),salary Number(9,2),userid varchar(25));
```

OUTPUT:



The screenshot shows the Oracle APEX SQL Workshop interface. In the top navigation bar, 'SQL Commands' is selected. The main area displays the SQL command for creating the 'MY_EMP' table:

```
1 CREATE TABLE MY_EMP(
2   ID NUMBER(4),
3   FIRST_NAME VARCHAR(25),
4   LAST_NAME VARCHAR(25),
5   SALARY NUMBER(9,2),
6   USERID VARCHAR(25)
7 );
```

In the bottom results pane, the output shows:

```
Table created.  
0.04 seconds
```

The status bar at the bottom right indicates the date and time as 23-02-2024, 10:34.

2.Add the first and second rows data to MY_EMPLOYEE table from the following sample data.

ID	Last_name	First_name	Userid	salary
1	Patel	Ralph	rpatel	895
2	Dancs	Betty	bdancs	860
3	Biri	Ben	bbiri	1100
4	Newman	Chad	Cnewman	750
5	Ropebur	Audrey	aropebur	1550

QUERY:

```
INSERT INTO MY_employee (id,last_name,first_name,userid,salary) VALUES('2','DANCS', 'BETTY', 'BDANCS', '860');
```

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. In the SQL Commands tab, the following SQL code is displayed:

```
1 CREATE TABLE MY_EMP(
2   ID NUMBER(4),
3   FIRST_NAME VARCHAR(25),
4   LAST_NAME VARCHAR(25),
5   SALARY NUMBER(9,2),
6   USERID VARCHAR(25)
7 );
8 INSERT INTO MY_EMP VALUES ('1','PATEL', 'RALPH', '895', 'RPATEL');
9 INSERT INTO MY_EMP VALUES ('2','DANCS', 'BETTY', '860', 'BDANCS');
10 INSERT INTO MY_EMP VALUES ('3','NEWMAN', 'CHAD', '750', 'CNEWMAN');
11 INSERT INTO MY_EMP VALUES ('4','BIRI', 'BEN', '1150', 'BBIRI');
```

The results section shows the output of the query:

```
1 row(s) inserted.
```

The status bar at the bottom indicates the session is 220701029@rajalakshmi.edu.in, the schema is arunmohan29, and the environment is Oracle APEX 23.2.4.

3. Display the table with values.

QUERY:

Select * from My_employee;

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. In the SQL Commands tab, the following SQL code is displayed:

```
1 CREATE TABLE MY_EMP(
2     ID NUMBER(4),
3     FIRST_NAME VARCHAR(25),
4     LAST_NAME VARCHAR(25),
5     SALARY NUMBER(9,2),
6     USERID VARCHAR(25)
7 );
8 INSERT INTO MY_EMP VALUES ('1','PATEL', 'RALPH', '895', 'RPATEL');
9 INSERT INTO MY_EMP VALUES ('2','DANCS', 'BETTY', '860', 'BDANCS');
10 INSERT INTO MY_EMP VALUES ('3','NEWMAN', 'CHAD', '750', 'CNEWMAN');
11 INSERT INTO MY_EMP VALUES ('4','BIRI', 'BEN', '1150', 'BBIRI');
12 SELECT * FROM MY_EMP
```

The Results tab displays the data inserted into the table:

ID	FIRST_NAME	LAST_NAME	SALARY	USERID
4	BIRI	BEN	1150	BBIRI
1	PATEL	RALPH	895	RPATEL
2	DANCS	BETTY	860	BDANCS
2	DANCS	BETTY	860	BDANCS
3	NEWMAN	CHAD	750	CNEWMAN

At the bottom, the status bar shows "5 rows returned in 0.01 seconds". The system status bar indicates "30°C Partly sunny" and the date "23-02-2024".

4.Populate the next two rows of data from the sample data. Concatenate the first letter of the first_name with the first seven characters of the last_name to produce Userid.

QUERY:

```
Insert into my_employee values('4','newan','chad','cewman',750');
```

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. In the SQL Commands tab, the following SQL code is entered:

```
1 CREATE TABLE MY_EMP(
2     ID NUMBER(4),
3     FIRST_NAME VARCHAR(25),
4     LAST_NAME VARCHAR(25),
5     SALARY NUMBER(9,2),
6     USERID VARCHAR(25)
7 );
8 INSERT INTO MY_EMP VALUES ('1','PATEL', 'RALPH', '895', 'RPATEL');
9 INSERT INTO MY_EMP VALUES ('2','DANCS', 'BETTY', '860', 'BDANCS');
```

The code creates a table named MY_EMP with columns ID, FIRST_NAME, LAST_NAME, SALARY, and USERID. It then inserts two rows of data into the table. The results section shows the message "1 row(s) inserted." and "0.01 seconds".

5. Make the data additions permanent.

QUERY:

Select * from My_employee;

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop, Team Development, and Gallery. The right side of the header shows the user's name, Arun Mohan29, and session information. The main area is titled "SQL Commands". The code editor contains the following SQL script:

```
1 CREATE TABLE MY_EMP(
2   ID NUMBER(4),
3   FIRST_NAME VARCHAR(25),
4   LAST_NAME VARCHAR(25),
5   SALARY NUMBER(9,2),
6   USERID VARCHAR(25)
7 );
8 INSERT INTO MY_EMP VALUES ('1','PATEL', 'RALPH', '895', 'RPATEL');
9 INSERT INTO MY_EMP VALUES ('2','DANCS', 'BETTY', '860', 'BDANCS');
10 INSERT INTO MY_EMP VALUES ('3','NEWMAN', 'CHAD', '750', 'CNEWMAN');
11 INSERT INTO MY_EMP VALUES ('4','BIRI', 'BEN', '1150', 'BBIRI');
12 SELECT * FROM MY_EMP
```

The results section displays the data inserted into the MY_EMP table:

ID	FIRST_NAME	LAST_NAME	SALARY	USERID
4	BIRI	BEN	1150	BBIRI
1	PATEL	RALPH	895	RPATEL
2	DANCS	BETTY	860	BDANCS
2	DANCS	BETTY	860	BDANCS
3	NEWMAN	CHAD	750	CNEWMAN

At the bottom, the status bar shows "5 rows returned in 0.01 seconds" and "Download" options. The taskbar at the very bottom includes icons for search, file, and system notifications.

6.Change the last name of employee 3 to Drexler.

QUERY:

```
Update My_employee Set last_name = 'DERXLER' where ID = 3;
```

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop, Team Development, and Gallery. The user is logged in as 'arunmohan29'. The main area is titled 'SQL Commands' and contains the following SQL code:

```
1 UPDATE MY_EMP
2 SET LAST_NAME = 'DREXLER' WHERE ID = 3;
3
4
5
6
7
```

Below the code, the 'Results' tab is selected, showing the output of the query:

```
1 row(s) updated.

0.01 seconds
```

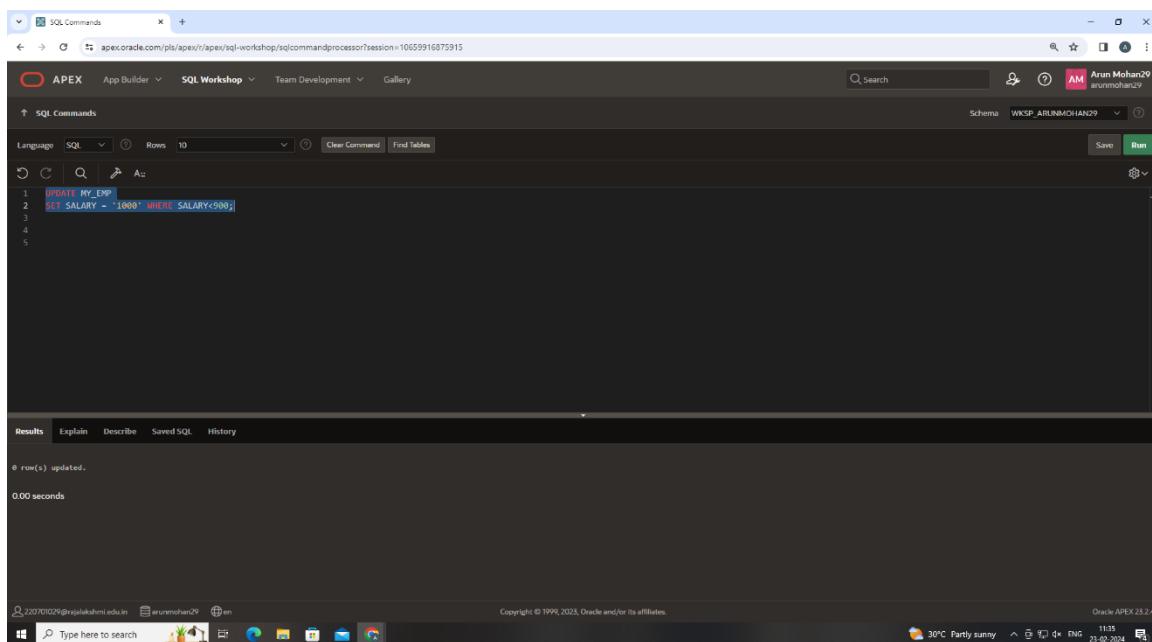
The bottom status bar displays the user's session information (220701029@njalakshmi.edu.in), the schema (arunmohan29), the operating system (Windows 10), and the system status (30°C, Partly sunny, 11:31, 23-02-2024).

7.Change the salary to 1000 for all the employees with a salary less than 900.

QUERY:

Update my_employee Set salary = 1000 where salary<900;

OUTPUT:



The screenshot shows the Oracle APEX SQL Workshop interface. In the SQL Commands pane, the following SQL code is entered:

```
1 UPDATE MY_EMP
2 SET SALARY = '1000' WHERE SALARY<900;
3
4
5
```

The Results tab shows the output of the query:

```
0 row(s) updated.
0.00 seconds
```

The status bar at the bottom indicates the session ID (220701029@replikashmi.edu.in), the schema (arunmohan29), the Oracle APEX version (23.2.4), the date (23-02-2024), and the weather (30°C Partly sunny).

8.Delete Betty dancs from MY _EMPLOYEE table.

QUERY:

Delete from My_employee where last_name= 'dancs';

OUTPUT:

The screenshot shows a SQL command window interface. At the top, there's a toolbar with various icons and buttons. Below the toolbar, the main area contains a command line with the following text:

```
1 delete from my_employee where last_name='dancs';
```

Below the command line, there's a results section. The first row of the results table displays the message "1 row(s) deleted." followed by the execution time "0.01 seconds".

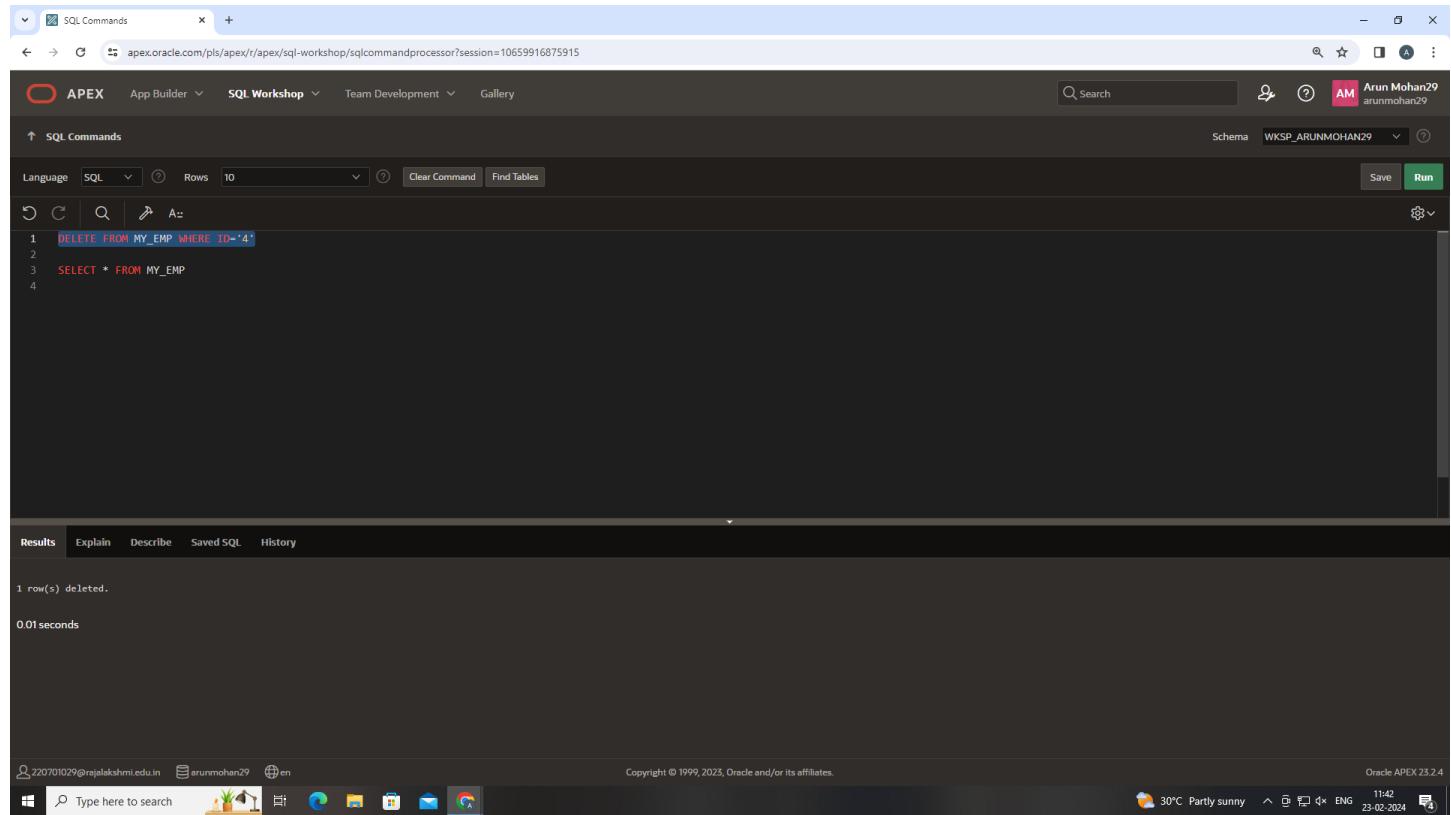
Results	Explain	Describe	Saved SQL	History
1 row(s) deleted. 0.01 seconds				

9.Empty the fourth row of the emp table.

QUERY:

Delete from emp where ID = '4';

OUTPUT:



The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop, Team Development, and Gallery. The user is logged in as 'AM arunmohan29'. The main workspace is titled 'SQL Commands' and contains the following SQL code:

```
1 DELETE FROM MY_EMP WHERE ID='4'
2
3 SELECT * FROM MY_EMP
4
```

The 'Run' button is visible at the bottom right of the code editor. Below the editor, the results tab is selected, showing the output of the executed command:

1 row(s) deleted.
0.01 seconds

The status bar at the bottom displays the user's email (220701029@njitlakshmi.edu.in), the schema (arunmohan29), the copyright notice (Copyright © 1999, 2023, Oracle and/or its affiliates.), and the system information (Oracle APEX 23.2.0).

Evaluation Procedure	Marks awarded
Query(5)	
Execution (5)	
Viva(5)	
Total (15)	
Faculty Signature	

RESULT:

INCLUDING CONSTRAINTS

EX_NO:3

DATE:

1. Add a table-level PRIMARY KEY constraint to the EMP table on the ID column. The constraint should be named at creation. Name the constraint my_emp_id_pk.

QUERY:

```
Alter table emp Add constraint My_dept_id_pk Primary key(ID);
```

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop, Team Development, and Gallery. The user is signed in as tif5. The main workspace is titled "SQL Commands". The command entered is:

```
1 ALTER TABLE DEPT ADD CONSTRAINT MY_DEPT_ID_PK PRIMARY KEY(ID);
```

The results section shows the output of the command:

```
Table altered.  
0.08 seconds
```

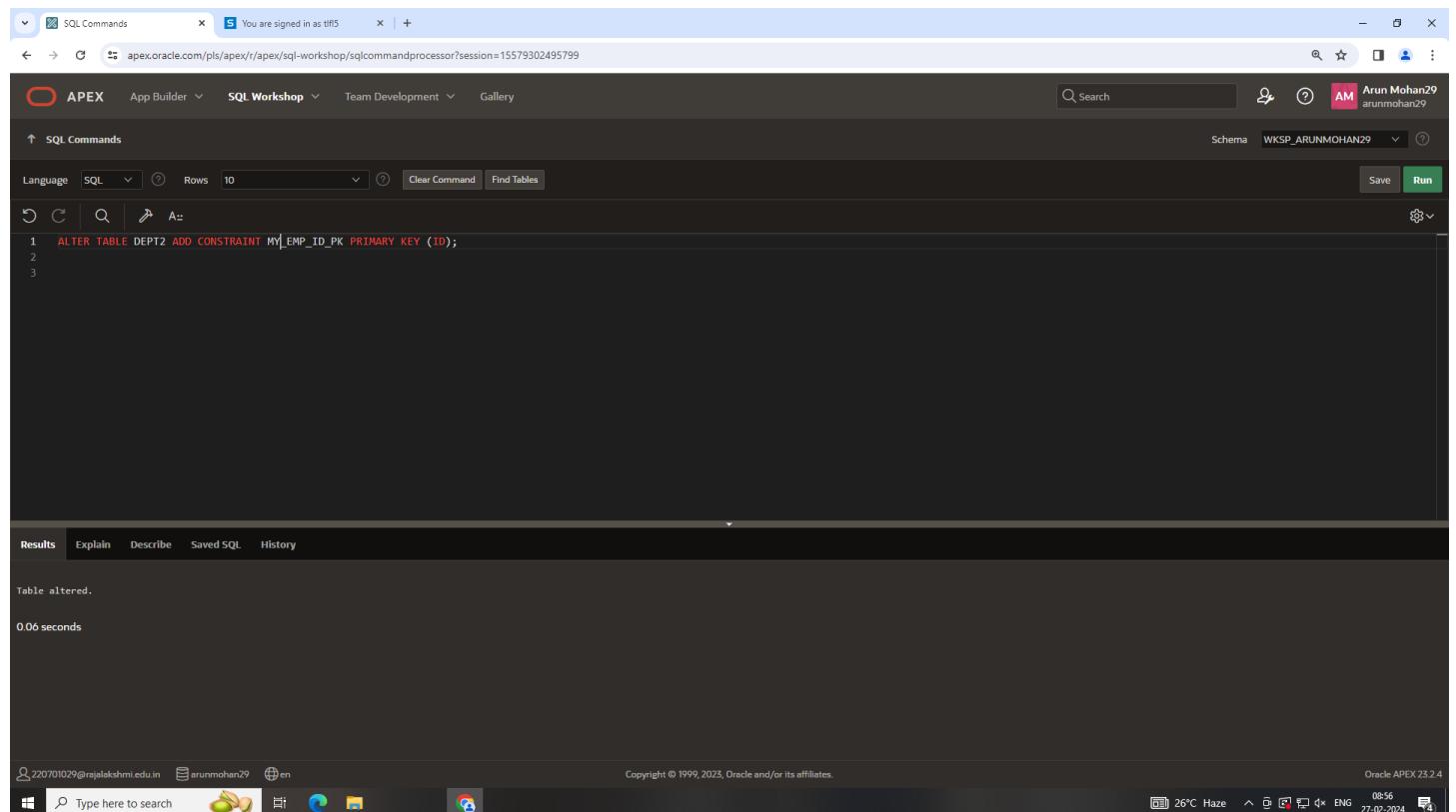
At the bottom, the status bar displays the user's session information and the Oracle APEX version (23.2.4).

2.Create a PRIMARY KEY constraint to the DEPT table using the ID column. The constraint should be named at creation. Name the constraint my_dept_id_pk.

QUERY:

```
Alter Table Dept Add constraint My_Emp_id_pk Primary key(ID);
```

OUTPUT:



The screenshot shows the Oracle APEX SQL Workshop interface. In the SQL Commands tab, the following SQL code is entered:

```
1 ALTER TABLE DEPT2 ADD CONSTRAINT MY_EMP_ID_PK PRIMARY KEY (ID);
2
3
```

The 'Run' button is highlighted in green. Below the code, the results show:

Table altered.
0.06 seconds

At the bottom, the Windows taskbar is visible with various icons and the system tray showing the date and time.

3.Add a column DEPT_ID to the EMP table. Add a foreign key reference on the EMP table that ensures that the employee is not assigned to nonexistent department. Name the constraint my_emp_dept_id_fk.

QUERY:

```
Alter Table Dept add constraint My_emp_dept_id_fk foreign key(ID) References Dept2(ID);
```

OUTPUT:

The screenshot shows a browser window with multiple tabs open. The active tab is 'oracle apex - Search' which displays the Oracle APEX SQL Workshop interface. The SQL Commands panel shows the following command:

```
1  ALTER TABLE DEPT ADD CONSTRAINT MY_EMP_DEPT_ID_FK FOREIGN KEY(ID) REFERENCES DEPT2(ID);
```

The 'Results' tab is selected, showing the output of the command:

```
Table altered.  
0.05 seconds
```

The status bar at the bottom left shows the session ID '220701029@rajalakshmi.edu.in' and the user 'arunmohan29'. The status bar at the bottom right shows the copyright notice 'Copyright © 1999, 2023, Oracle and/or its affiliates.' and the version 'Oracle APEX 23.2.4'.

4.Modify the EMP table. Add a COMMISSION column of NUMBER data type, precision 2, scale 2. Add a constraint to the commission column that ensures that a commission value is greater than zero.

QUERY:

```
Alter Table Dept2 Add(commission number(2,2),constraint Cn check(commission));
```

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. In the top navigation bar, the 'SQL Commands' tab is selected. The main area contains the following SQL command:

```
1 ALTER TABLE DEPT2 ADD(commission number(2,2),constraint Cn check(commission>0));
```

Below the command, the 'Results' tab is active, showing the output of the executed query:

```
Table altered.
```

Execution details are shown at the bottom of the results pane:

```
0.07 seconds
```

The status bar at the bottom of the browser window displays the user information and system status:

```
220701029@njalakshmi.edu.in arunmohan29 en Copyright © 1999, 2025, Oracle and/or its affiliates. Oracle APEX 23.2.4 26°C Haze 09:16 27-02-2024
```

Evaluation Procedure	Marks awarded
Query(5)	
Execution (5)	
Viva(5)	
Total (15)	
Faculty Signature	

RESULT:

Writing Basic SQL SELECT Statements

EX_NO:4

DATE:

- 1.The following statement executes successfully.

Identify the Errors

```
SELECT employee_id, last_name  
sal*12 ANNUAL SALARY  
FROM employees;
```

QUERY:

- 2.Show the structure of departments the table. Select all the data from it.

QUERY:

OUTPUT:

- 3.Create a query to display the last name, job code, hire date, and employee number for each employee, with employee number appearing first.

QUERY:

OUTPUT:

4. Provide an alias STARTDATE for the hire date.

QUERY:

OUTPUT:

5.Create a query to display unique job codes from the employee table.

QUERY:

OUTPUT:

6.Display the last name concatenated with the job ID , separated by a comma and space, and name the column EMPLOYEE and TITLE.

QUERY:

OUTPUT:

7.Create a query to display all the data from the employees table. Separate each column by a comma. Name the column THE_OUTPUT.

QUERY:

OUTPUT:

Evaluation Procedure	Marks awarded
Query(5)	
Execution (5)	
Viva(5)	
Total (15)	
Faculty Signature	

RESULT:

RESTRICTING AND SORTING DATA

EX_NO:5

DATE:

1.Create a query to display the last name and salary of employees earning more than 12000.

QUERY:

Select last_name,salary from employees where salary>12000;

OUTPUT:

The screenshot shows a browser window for apex.oracle.com with the URL: apex.oracle.com/pls/apex/r/apex/sql-workshop/sqlcommandprocessor?session=12243960650244. The page title is "SQL Commands". The user is connected to the schema "WKSP_ARUNMOHAN29". The SQL command entered is: "Select last_name,salary from employees where salary>12000;". The results section displays the following data:

LAST_NAME	SALARY
Dhoni	20000
kolhi	14000
khan	80200
kumar	15000
jadeja	31004

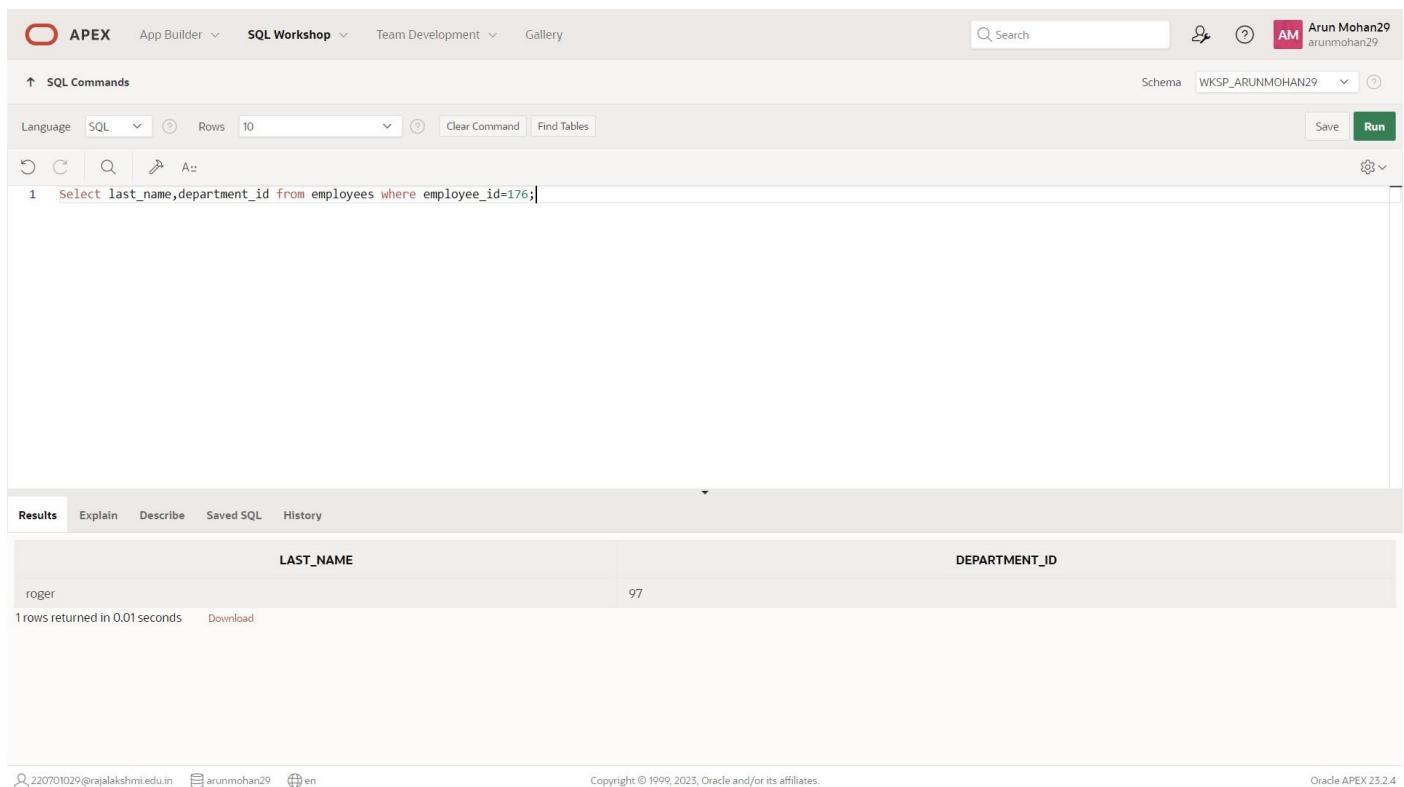
Below the table, it says "5 rows returned in 0.00 seconds". The bottom status bar shows system information: 220701029@rajalakshmi.edu.in, arunmohan29, en, Copyright © 1999, 2025, Oracle and/or its affiliates, Oracle APEX 23.2.4, 29°C Humid, Search, and various system icons.

2. Create a query to display the employee last name and department number for employee number 176.

QUERY:

```
Select last_name,department_id from employees where employee_id=176;
```

OUTPUT:



The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop, Team Development, and Gallery. On the right, there's a search bar, a user icon for 'Arun Mohan29', and a schema dropdown set to 'WKSP_ARUNMOHAN29'. The main workspace is titled 'SQL Commands' and contains a single line of SQL code: 'Select last_name,department_id from employees where employee_id=176;'. Below the code, the 'Results' tab is selected, displaying the output of the query. The results show one row with columns 'LAST_NAME' and 'DEPARTMENT_ID', both containing the value 'roger'. At the bottom left, it says '1 rows returned in 0.01 seconds'. The bottom right corner indicates the version 'Oracle APEX 23.2.4'.

LAST_NAME	DEPARTMENT_ID
roger	97

1 rows returned in 0.01 seconds Download

Copyright © 1999, 2023, Oracle and/or its affiliates.

Oracle APEX 23.2.4

3. Create a query to display the last name and salary of employees whose salary is not in the range of 5000 and 12000.
(hints: not between)

QUERY:

```
select last_name,salary from employees where salary not between 5000 and 12000;
```

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. A search bar and user profile 'Arun Mohan29' are also present. The main workspace displays the SQL command:

```
1 select last_name,salary from employees where salary not between 5000 and 12000;
```

Below the command, the results section shows a table with two columns: LAST_NAME and SALARY. The data returned is:

LAST_NAME	SALARY
Dhoni	20000
kolhi	14000
khan	80200
kumar	15000
jadeja	31004

At the bottom, it indicates "5 rows returned in 0.01 seconds".

4. Display the employee last name, job ID, and start date of employees hired between February 20,1998 and May 1,1998.order the query in ascending order by start date.(hints: between)

QUERY:

```
Select last_name,job_id,hire_date from employees where hire_date between '02/20/1998' and '05/01/1998';
```

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. On the right, there's a search bar, a user profile for 'Arun Mohan29', and a schema dropdown set to 'WKSP_ARUNMOHAN29'. The main workspace is titled 'SQL Commands' and contains a SQL editor with the following command:

```
1 Select last_name,job_id,hire_date from employees where hire_date between '02/20/1998' and '05/01/1998';
```

Below the editor, the 'Results' tab is selected, showing the output of the query:

LAST_NAME	JOB_ID	HIRE_DATE
nolan	244	03/21/1998

At the bottom of the results panel, it says '1 rows returned in 0.01 seconds' and has a 'Download' link. The footer of the page includes copyright information for Oracle and the APEX version.

5.Display the last name and department number of all employees in departments 20 and 50 in alphabetical order by name.(hints: in, orderby) **QUERY:** select last_name,department_id from employees where department_id in(20,50) order by last_name;

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop, Team Development, and Gallery. On the right, there's a search bar, user profile for 'Arun Mohan29', and a 'Run' button. The main workspace is titled 'SQL Commands'. It shows a SQL command being run:

```
1 select last_name,department_id from employees where department_id in(20,50) order by last_name;
```

The results tab is selected, displaying the output of the query:

LAST_NAME	DEPARTMENT_ID
nolan	50

Below the table, it says '1 rows returned in 0.00 seconds' and has a 'Download' link.

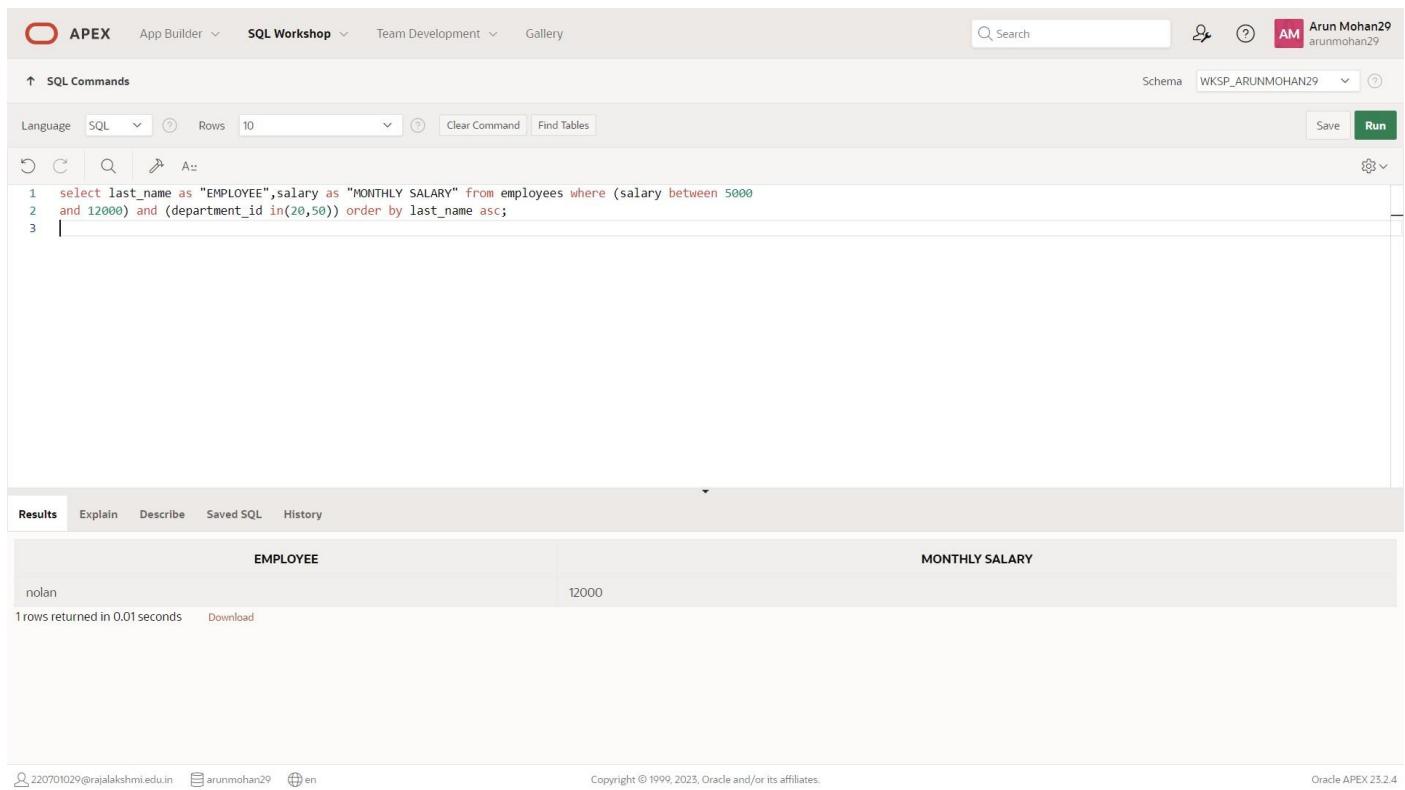
At the bottom, the footer includes links for 220701029@rajalakshmi.edu.in, arunmohan29, and en, along with copyright information: Copyright © 1999, 2023, Oracle and/or its affiliates. and Oracle APEX 23.2.4.

6. Display the last name and salary of all employees who earn between 5000 and 12000 and are in departments 20 and 50 in alphabetical order by name. Label the columns EMPLOYEE, MONTHLY SALARY respectively.(hints: between, in)

QUERY:

```
select last_name as "EMPLOYEE", salary as "MONTHLY SALARY" from employees where (salary between 5000 and 12000)  
and (department_id in(20,50)) order by last_name asc;
```

OUTPUT:



The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes 'APEX', 'App Builder', 'SQL Workshop' (selected), 'Team Development', and 'Gallery'. On the right, there's a search bar, a user icon for 'Arun Mohan29', and a schema dropdown set to 'WKSP_ARUNMOHAN29'. The main workspace is titled 'SQL Commands' and contains the following SQL code:

```
1 select last_name as "EMPLOYEE", salary as "MONTHLY SALARY" from employees where (salary between 5000  
2 and 12000) and (department_id in(20,50)) order by last_name asc;  
3 |
```

Below the code, the 'Results' tab is selected, showing the output:

EMPLOYEE	MONTHLY SALARY
nolan	12000

1 rows returned in 0.01 seconds [Download](#)

At the bottom, the footer includes links for '220701029@rajalakshmi.edu.in', 'arunmohan29', and 'en'. It also states 'Copyright © 1999, 2023, Oracle and/or its affiliates.' and 'Oracle APEX 23.2.4'.

7.Display the last name and hire date of every employee who was hired in 1994.(hints: like)

QUERY:

```
select last_name,hire_date from employees where hire_date like '%1994';
```

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. On the right, it shows the user 'Arun Mohan29' and their schema 'WKSP_ARUNMOHAN29'. The main area is titled 'SQL Commands' and contains a code editor with the following SQL statement:

```
1 select last_name,hire_date from employees where hire_date like '%1994';
2
```

Below the code editor, there are tabs for Results, Explain, Describe, Saved SQL, and History. The Results tab is selected, displaying the query results in a table:

LAST_NAME	HIRE_DATE
roger	01/01/1994
kumar	02/05/1994

Below the table, it says '2 rows returned in 0.00 seconds' and has a 'Download' link.

At the bottom of the page, there are footer links for 220701029@rajalakshmi.edu.in, arunmohan29, and en. The copyright notice reads 'Copyright © 1999, 2023, Oracle and/or its affiliates.' and the page version is 'Oracle APEX 23.2.4'.

8.Display the last name and job title of all employees who do not have a manager.(hints: is null)

QUERY:

```
select last_name,job_id from employees where manager_id is null;
```

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. On the right, there's a search bar, user information for 'Arun Mohan29' (arumohan29), and a 'Run' button. The main workspace is titled 'SQL Commands'. It contains a language selector set to 'SQL', a row limit of '10', and buttons for Clear Command and Find Tables. Below this is a code editor with the following content:

```
1 select last_name,job_id from employees where manager_id is null;
2
```

At the bottom, the results tab is selected, showing a single row of data:

LAST_NAME	JOB_ID
miller	6

Below the results, it says '1 rows returned in 0.01 seconds' and has a 'Download' link. The footer includes copyright information for Oracle and the APEX version.

9. Display the last name, salary, and commission for all employees who earn commissions. Sort data in descending order of salary and commissions.(hints: is not null, order by)

QUERY:

```
select last_name,salary,commission_no from employees where commission_no is not null order by salary desc;
```

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop, Team Development, and Gallery. The right side shows the user profile "Arun Mohan29" and the schema "WKSP_ARUNMOHAN29". The main workspace is titled "SQL Commands" and contains the following SQL code:

```
1 select last_name,salary,commission_no from employees where commission_no is not null order by salary
2 desc;
3
4
```

The "Run" button is visible at the bottom right of the command input area. Below the command input, there are tabs for "Results", "Explain", "Describe", "Saved SQL", and "History". The "Results" tab is selected, displaying the query results in a grid format:

LAST_NAME	SALARY	COMMISSION_NO
khan	80200	.55
Dhoni	20000	.32
kumar	15000	.67
kolhi	14000	.98
nolan	12000	.89
roger	11000	.76

At the bottom of the page, there are footer links for "220701029@rajalakshmi.edu.in", "arunmohan29", and "en", along with copyright information "Copyright © 1999, 2023, Oracle and/or its affiliates." and the text "Oracle APEX 23.2.4".

10. Display the last name of all employees where the third letter of the name is *a*.(hints:like)

QUERY:

```
select last_name from employees where last_name like '__a%';
```

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. On the right, there's a search bar, user profile for Arun Mohan29, and a schema dropdown set to WKSP_ARUNMOHAN29. The main workspace is titled "SQL Commands". It features a toolbar with icons for refresh, search, and run, along with dropdowns for language (SQL selected) and rows (10). Below the toolbar is a text area containing the SQL query:

```
1 select last_name from employees where last_name like '__a%';  
2  
3
```

At the bottom of the workspace, there are tabs for Results, Explain, Describe, Saved SQL, and History. The Results tab is active, showing the output of the query:

LAST_NAME
khan

Below the results, it says "1 rows returned in 0.01 seconds" and has a "Download" link. The footer of the page includes email and profile links for the user, copyright information (Copyright © 1999, 2023, Oracle and/or its affiliates.), and the text "Oracle APEX 25.2.4".

11. Display the last name of all employees who have an a and an e in their last name.(hints: like)

QUERY:

```
select last_name from employees where last_name like '%a%' and last_name like '%e%';
```

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. On the right, there's a search bar, user profile, and session information (AM Arun Mohan29). The main workspace is titled 'SQL Commands' and contains a code editor with the following SQL query:

```
1 select last_name from employees where last_name like '%a%' and last_name like '%e%';
2
3
```

Below the code editor, the results tab is selected, showing a single row of data:

LAST_NAME
jadeja

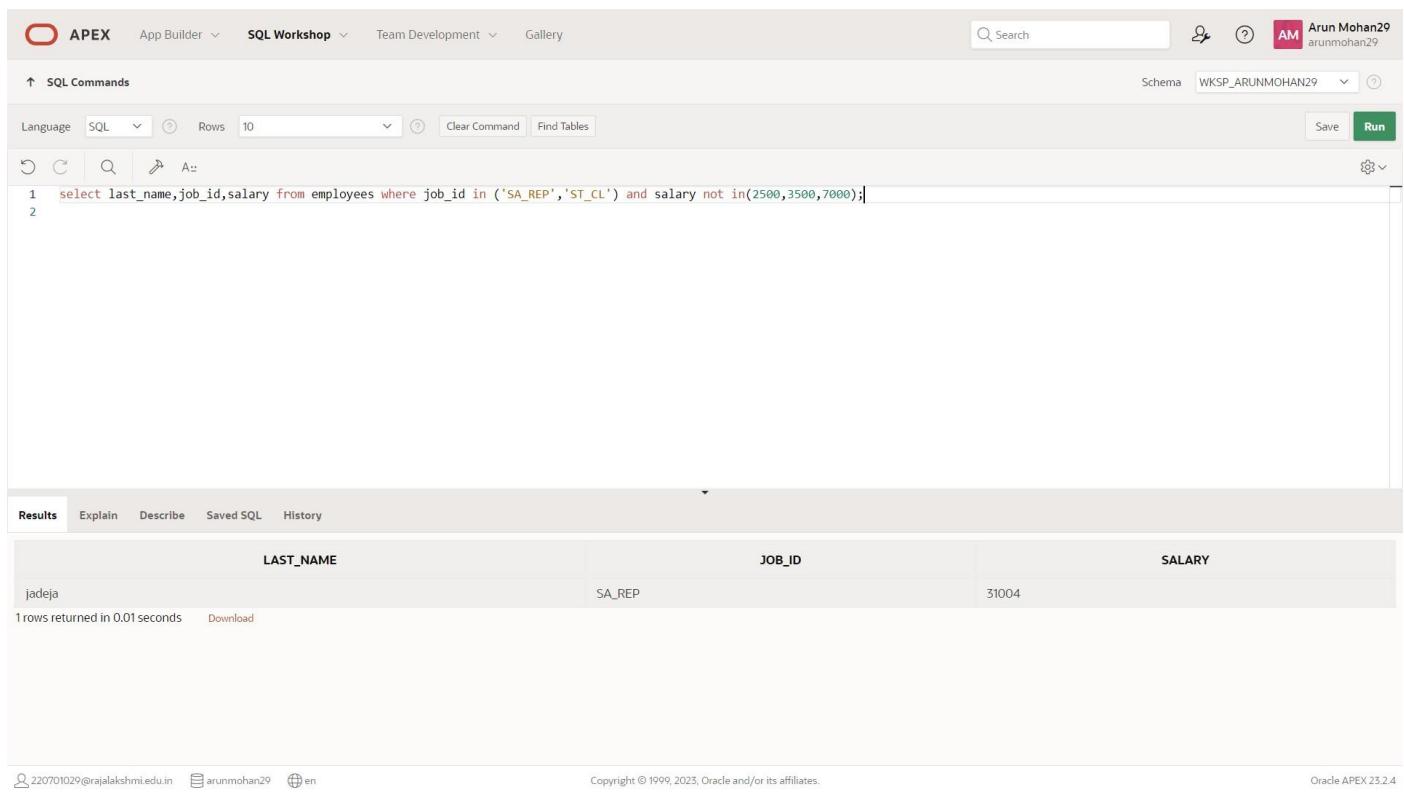
At the bottom, it indicates 1 row returned in 0.01 seconds and provides download options.

12. Display the last name and job and salary for all employees whose job is sales representative or stock clerk and whose salary is not equal to 2500 ,3500 or 7000.(hints:in,not in)

QUERY:

```
select last_name,job_id,salary from employees where job_id in ('SA_REP','ST_CL') and salary not in(2500,3500,7000);
```

OUTPUT:



The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. On the right, there's a search bar, user information for 'Arun Mohan29', and a schema dropdown set to 'WKSP_ARUNMOHAN29'. The main workspace is titled 'SQL Commands' and contains the following SQL code:

```
1 select last_name,job_id,salary from employees where job_id in ('SA_REP','ST_CL') and salary not in(2500,3500,7000);
2
```

Below the code, the 'Results' tab is selected, showing the output of the query:

LAST_NAME	JOB_ID	SALARY
jadeja	SA_REP	31004

At the bottom of the results pane, it says '1 rows returned in 0.01 seconds' and has a 'Download' link. The footer of the page includes copyright information for Oracle and the APEX version.

13. Display the last name, salary, and commission for all employees whose commission amount is 20%. (hints: use predicate logic)

QUERY:

```
select last_name,salary,commission_no from employees where commission_no=0.2;
```

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. On the right, there's a search bar, user profile, and session information for 'Arun Mohan29'. The main workspace is titled 'SQL Commands' and contains a command line with the query: 'select last_name,salary,commission_no from employees where commission_no=0.2;'. Below the command line, the 'Results' tab is selected, showing the message 'no data found'. At the bottom, footer information includes email and session details, and a copyright notice: 'Copyright © 1999, 2025, Oracle and/or its affiliates.' and 'Oracle APEX 23.2.4'.

Evaluation Procedure	Marks awarded
Query(5)	
Execution (5)	
Viva(5)	
Total (15)	
Faculty Signature	

RESULT:

SINGLE ROW FUNCTIONS

EX_NO:6

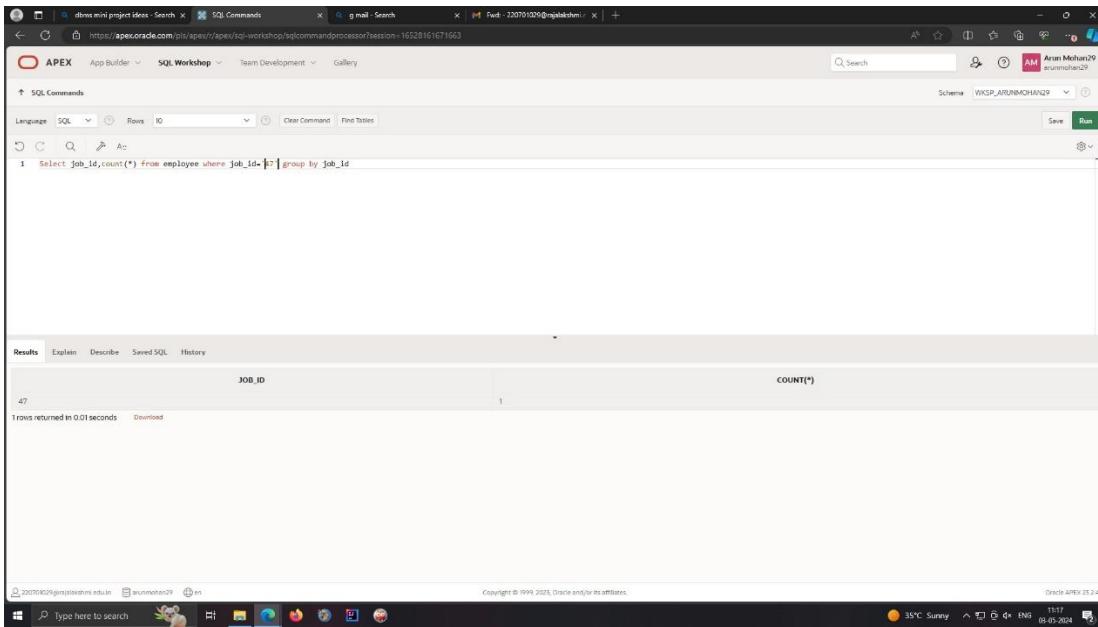
DATE:

1. Write a query to display the current date. Label the column Date.

QUERY:

```
select sysdate from dual;
```

OUTPUT:



The screenshot shows a browser window for the Oracle APEX SQL Workshop. The URL is https://apex.oracle.com/pls/apex/r/apex/sql-workshop/sqlcommandprocessor?session=16520161671563. The page title is "APEX - App Builder - Team Development - Gallery". The user is logged in as Arun Mohan29. The SQL Commands tab is selected. The SQL editor contains the following query:

```
1 Select job_id, count(*) #from employee where job_id=47 group by job_id
```

The results section shows the output of the query:

JOB_ID	COUNT(*)
47	1

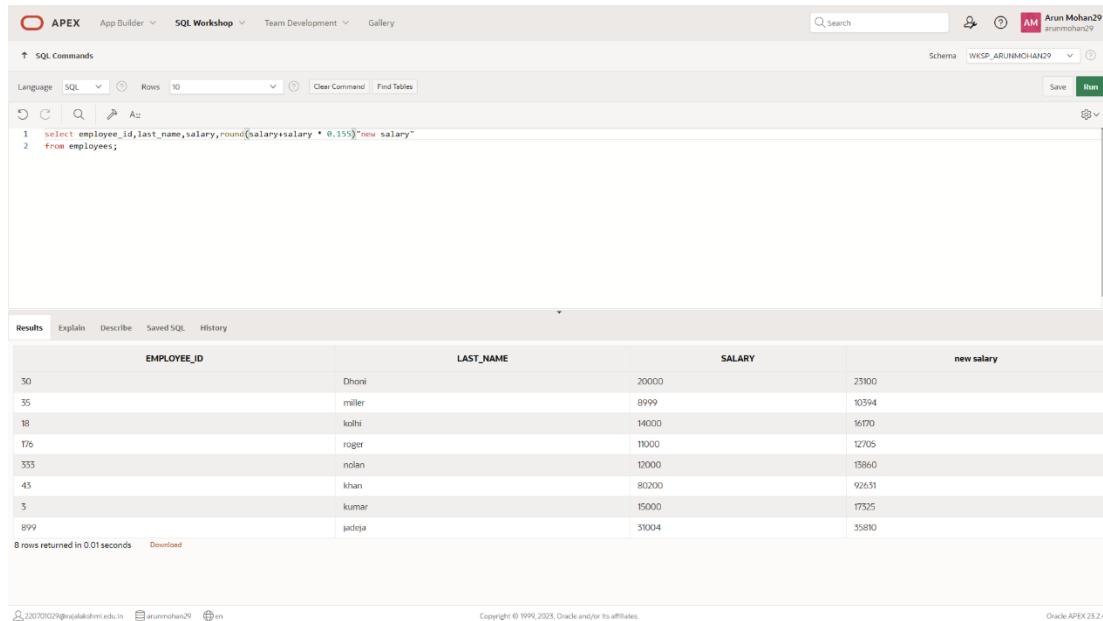
Below the table, it says "1 rows returned in 0.01 seconds" and "Download". The status bar at the bottom right indicates "Oracle APEX 23.1.2" and the system status "35°C Sunny 15:17 08-05-2024".

2. The HR department needs a report to display the employee number, last name, salary, and increased by 15.5% (expressed as a whole number) for each employee. Label the column New Salary.

QUERY:

```
select employee_id, last_name, salary, salary+(15.5/100*salary) "new_salary" from empo21;
```

OUTPUT



The screenshot shows the Oracle APEX SQL Workshop interface. The SQL command entered is:

```
1 select employee_id, last_name, salary, round(salary+salary * 0.155) "new salary"
2 from employees;
```

The results section displays the following data:

EMPLOYEE_ID	LAST_NAME	SALARY	new salary
30	Dhoni	20000	23100
35	miller	8999	10394
18	kohli	14000	16170
176	roger	11000	12705
333	nolan	12000	13860
45	khan	80200	92631
5	kumar	15000	17325
899	jadeja	31004	35810

8 rows returned in 0.01 seconds [Download](#)

OUTPUT:

3. Modify your query lab_03_02.sql to add a column that subtracts the old salary from the new salary.
Label the column Increase.

QUERY:

```
select employee_id, last_name, salary, salary+(15.5/100*salary) "new_salary", new_salary-salary as  
"Increase" from empo21;
```

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop, Team Development, and Gallery. The user is logged in as 'Arun Mohan29'. The schema selected is 'WKSP_ARUNMOHAN29'. The main area displays a SQL command:

```
1 select to_char(sysdate,'mm-dd-yy') as "date"  
2 from dual;
```

The results tab shows the output:

date
05-12-24

1 rows returned in 0.02 seconds

4. Write a query that displays the last name (with the first letter uppercase and all other letters lowercase) and the length of the last name for all empo21 whose name starts with the letters J, A, or M. Give each column an appropriate label. Sort the results by the empo21' last names.

QUERY:

```
select initcap(last_name), length(last_name) as "Length_of_last_name" from empo21 where last_name like  
'J%' or last_name like 'A%' or last_name like 'M%' order by last_name asc;
```

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop, Team Development, and Gallery. The user is logged in as 'Arun Mohan29'. The schema selected is 'WKSP_ARUNMOHAN29'. The main area displays a SQL command:

```
1 select to_char(sysdate,'mm-dd-yy') as "date"  
2 from dual;
```

The results tab shows the output:

date
05-12-24

1 rows returned in 0.02 seconds

5. Rewrite the query so that the user is prompted to enter a letter that starts the last name. For example, if the user enters H when prompted for a letter, then the output should show all empo21 whose last name starts with the letter H.

QUERY:

```
select initcap(last_name),length(last_name) as "Length_of_last_name" from empo21 where last_name like 'J%' or last_name like 'A%' or last_name like 'M%' order by last_name ;
```

OUTPUT:

LAST_NAME	JOB_ID	DEPT_NAME	SALARY	GRADE_LEVEL
askothi	8	CSE	14000	4th
miller	6	CSOS	8999	5th
Phoni	07	CSBS	20000	3rd

6. The HR department wants to find the length of employment for each employee. For each employee, display the last name and calculate the number of months between today and the date on which the employee was hired. Label the column MONTHS_WORKED. Order your results by the number of months employed. Round the number of months up to the closest whole number.

QUERY:

```
select last_name,round((sysdate-hire_date)/30,0) as "MONTHS_WORKED" from empo21 order by round((sysdate-hire_date)/30,0) asc;
```

OUTPUT:

JOB_ID	LOCATION_ID
8	5

7. Create a report that produces the following for each employee:

<employee last name> earns <salary> monthly but wants <3 times salary>. Label the column Dream Salaries.

QUERY:

```
select last_name||' earns'||salary||' monthly but wants'||salary*3 as "DREAM_SALARIES" from empo21;
```

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. The SQL Commands tab contains the following code:

```
1 Select e.last_name,e.department_id,d.dept_name,d.loc_id,l.city From employee e,mydept d,location l
2 where e.department_id=d.deptid and d.loc_id=l.locationid and e.commission_no is not null;
```

The Results tab displays the output of the query:

LAST_NAME	DEPARTMENT_ID	DEPT_NAME	LOC_ID	CITY
kohli	80	CSE	1	Chennai
kohli	80	CSE	1	Chennai

2 rows returned in 0.02 seconds

8. Create a query to display the last name and salary for all empo21. Format the salary to be 15 characters long, left-padded with the \$ symbol. Label the column SALARY.

QUERY:

```
select last_name,lpad(salary,15,'$') as "SALARY" from empo21;
```

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. The SQL Commands tab contains the following code:

```
1 Select employee.last_name,mydept.dept_name From employee,mydept
2 where employee.department_id=mydept.deptid and last_name like 'kak'
```

The Results tab displays the output of the query:

LAST_NAME	DEPT_NAME
akkohli	CSE

1 rows returned in 0.00 seconds

9. Display each employee's last name, hire date, and salary review date, which is the first Monday after six months of service. Label the column REVIEW. Format the dates to appear in the format similar to "Monday, the Thirty-First of July, 2000."

QUERY:

```
SELECT last_name,hire_date,TO_CHAR(NEXT_DAY(ADD_MONTHS(hire_date, 6),  
'MONDAY'),'FMDay, "the "FMDD "of "FMMonth, YYYY') AS REVIEW FROM empo21;
```

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. The SQL Commands tab contains the following code:

```
1 select e.last_name,e.department_id,e.job_id,d.dept_name from employee e join mydept d  
2 on(e.department_id=d.deptid) join location on (d.loc_id=location.locationid) where  
3 lower(location.city)='toronto';
```

The Results tab displays the output:

LAST_NAME	DEPARTMENT_ID	JOB_ID	DEPT_NAME
Dhoni	79	07	CSBS

1 rows returned in 0.01 seconds Download

Copyright © 1996, 2023, Oracle and/or its affiliates. Oracle APEX 23.2.4

10. Display the last name, hire date, and day of the week on which the employee started. Label the column DAY. Order the results by the day of the week, starting with Monday.

QUERY:

```
SELECT last_name,hire_date,TO_CHAR(hire_date,'Day') as Day from empo21 order by  
TO_CHAR(hire_date,'Day');
```

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. The SQL Commands tab contains the following code:

```
1 select e.last_name,e.manager_name,e.dept_name,d.loc_id,d.loc,e.salary from employee e join  
2 mydept d on e.employee_id=d.empid order by e.employee_id
```

The Results tab displays the output:

LAST_NAME	MANAGER_NAME	DEPT_NAME	LOC_ID	LOC	SALARY
aakothi	vishwa	ECE	1	Chennai	14000

1 rows returned in 0.02 seconds Download

Copyright © 1996, 2023, Oracle and/or its affiliates. Oracle APEX 23.2.4

Evaluation Procedure	Marks awarded
Query(5)	
Execution (5)	
Viva(5)	
Total (15)	
Faculty Signature	

RESULT:

DISPLAYING DATA FROM MULTIPLE TABLES

EX_NO:7

DATE:

1. Write a query to display the last name, department number, and department name for all employees.

QUERY:

```
Select e.last_name,e.department_number,d.dept_id from empo21 e,dept23 d where  
e.department_number=d.dept_id;
```

The screenshot shows the Oracle APEX SQL Workshop interface. The query window contains the following SQL code:

```
1 SELECT e.last_name, e.job_id, d.dept_name, e.salary, j.grade_level FROM employee e  
2 JOIN mydept d ON (e.department_id = d.dept_id) JOIN job_grade j ON (e.salary BETWEEN  
3 j.lower_bound AND j.high_bound);  
4
```

The results window displays the output of the query:

LAST_NAME	JOB_ID	DEPT_NAME	SALARY	GRADE_LEVEL
aaakothi	8	CSE	14000	4th
miller	6	CSE	8999	5th
Dhoni	07	CSE	20000	3rd

3 rows returned in 0.06 seconds

OUTPUT:

2. Create a unique listing of all jobs that are in department 80. Include the location of the department in the output.

QUERY:

```
select distinct job_id,loc_id from empo21 e,dept23 d where e.department_number=d.dept_id and
```

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. The query window contains the following SQL code:

```
1 select e.department_id "Dept",e.last_name "colleague" From employee e Join employee c on  
2 (e.department_id=c.department_id) where e.employee_id > c.employee_id order by  
3 e.department_id,e.last_name..last_name;
```

The results window displays the output of the query:

Dept	colleague
79	Dhoni
79	miller

2 rows returned in 0.01 seconds

3. Write a query to display the employee last name, department name, location ID, and city of all employees who earn a commission

QUERY:

```
Select e.last_name,e.department_number,d.dept_name,d.loc_id,l.city from emp021 e,dept23 d,location l where e.department_number=d.dept_id and d.loc_id=l.location_id and e.commission_pct is not null;
```

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. The SQL command is:

```
1 select last_name "Employee",employee_id "emp#",manager_name "manager",manager_id "mgr#"
2  from employee
```

The results table displays the following data:

Employee	emp#	manager	Mgr #
Dhoni	30	king	8
miller	35	bhar	-
aakohli	58	vishwa	6
Hammer	170	king	89
pandya	333	king	77
ikhans	45	-	2
kumar	3	-	2
Jadeja	899	-	24

0 rows returned in 0.01 seconds

4. Display the employee last name and department name for all employees who have an a(lowercase).

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. The SQL command is:

```
1 select e.last_name,e.manager_name,e.dept_name,d.loc_id,d.loc.e.salary from employee e join
2 mydept d on e.employee_id=d.empid order by e.employee_id
```

The results table displays the following data:

LAST_NAME	MANAGER_NAME	DEPT_NAME	LOC_ID	LOC	SALARY
aakohli	vishwa	ECE	1	Chennai	14000

1 rows returned in 0.02 seconds

5. Write a query to display the last name, job, department number, and department name for all employees who work in Toronto.

QUERY:

```
Select e.last_name,e.department_number,e.job_id,d.dept_name from empo21 e join dept d  
on(e.department_number=d.dept_id) join location on (d.location_id=location.location_id) where  
lower(location.city)=’toronto’;
```

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. The SQL command entered is:

```
1 select e.last_name,e.department_id,e.job_id,d.dept_name from employee e join mydept d  
2 on(e.department_id=d.deptid) join location on (d.location_id=location.locationid) where  
3 lower(location.city)=’toronto’;
```

The results section displays the following data:

LAST_NAME	DEPARTMENT_ID	JOB_ID	DEPT_NAME
Dhoni	79	07	CSBS

1 rows returned in 0.01 seconds [Download](#)

6. Display the employee last name and employee number along with their manager’s last name and manager number. Label the columns Employee, Emp#, Manager, and Mgr#, Respectively

QUERY:

```
Select w.last_name “Employee”,w.emp_id “emp#”,m.last_name ‘manager’,m.emp_id “Mgr#” from empo21 m  
on (w.manager_id=m.emp_id);
```

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. The SQL command entered is:

```
1 select e.last_name,e.department_id,e.job_id,d.dept_name from employee e join mydept d  
2 on(e.department_id=d.deptid) join location on (d.location_id=location.locationid) where  
3 lower(location.city)=’toronto’;
```

The results section displays the following data:

LAST_NAME	DEPARTMENT_ID	JOB_ID	DEPT_NAME
Dhoni	79	07	CSBS

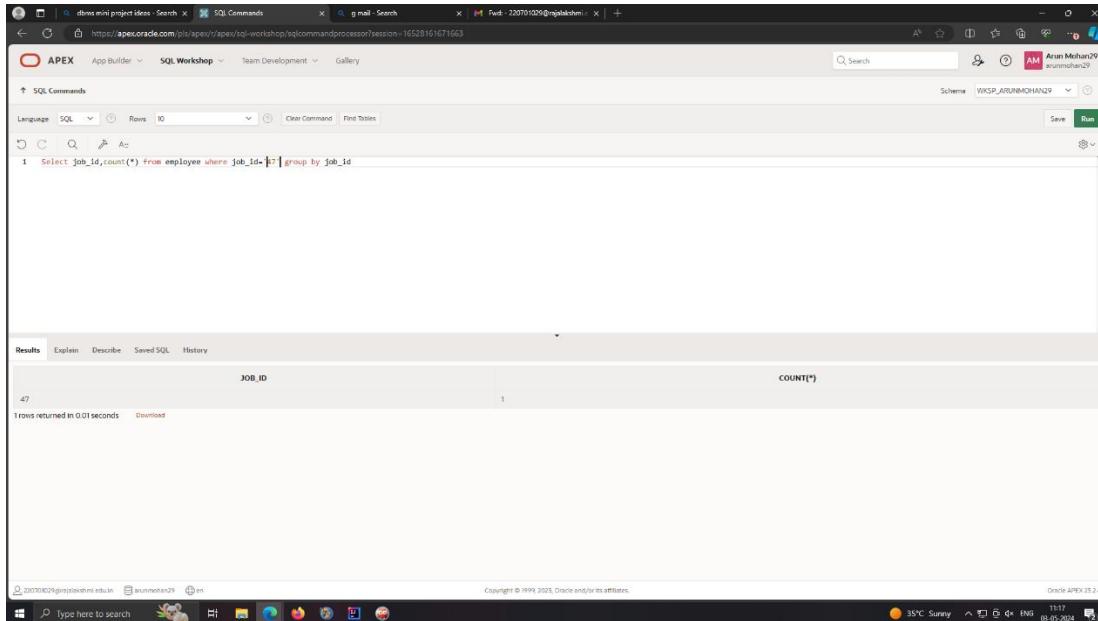
1 rows returned in 0.01 seconds [Download](#)

7. Modify lab4_6.sql to display all employees including King, who has no manager. Order the results by the employee number.

QUERY:

Select w.last_name "Employee",w.emp_id "emp#",m.last_name 'manager',m.emp_id "Mgr#" from empo21 w left outer join empo21 m on (w.manager_id=m.emp_id);

OUTPUT:



The screenshot shows the Oracle APEX SQL Workshop interface. The query entered is:

```
1 Select job_id, count(*) from employee where job_id=47 group by job_id
```

The results table shows one row:

JOB_ID	COUNT(*)
47	1

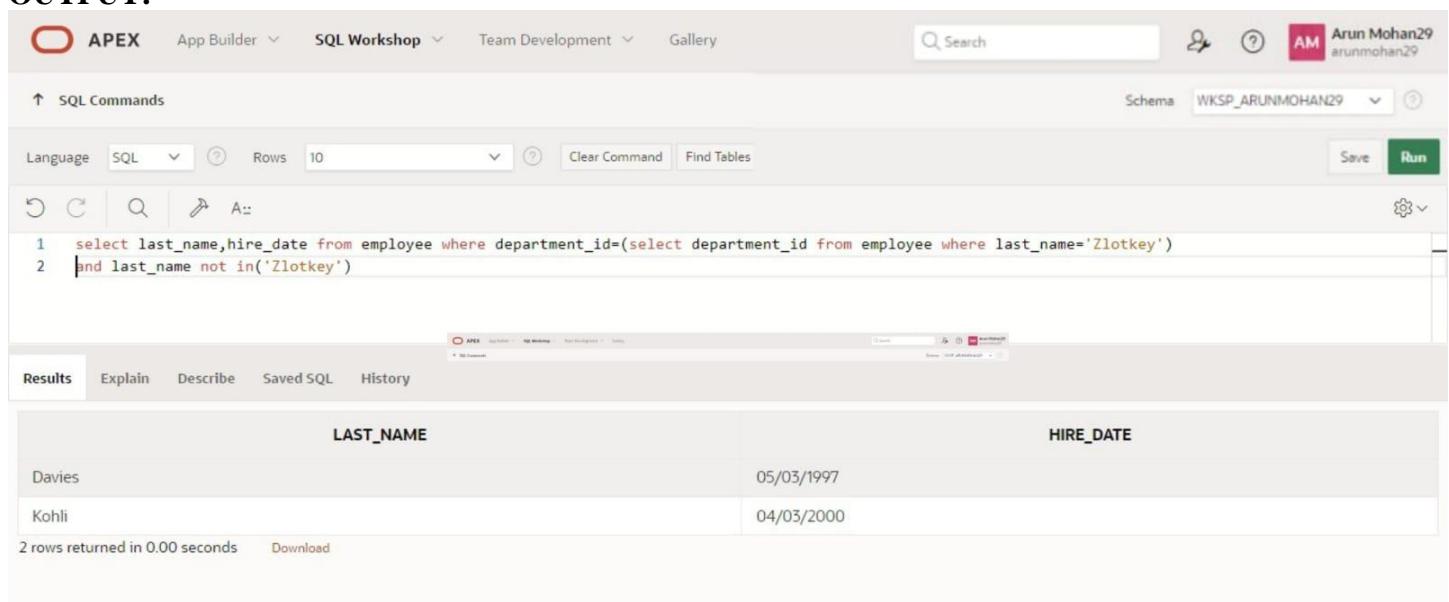
1 rows returned in 0.01 seconds

8. Create a query that displays employee last names, department numbers, and all the employees who work in the same department as a given employee. Give each column an appropriate label

QUERY:

select e.department_number dept23,e.last_name colleague from empo21 e join empo21 c on (e.department_number=c.department_number) where e.emp_id <> c.emp_id order by e.department_number,e.last_name,c.last_name;

OUTPUT:



The screenshot shows the Oracle APEX SQL Workshop interface. The query entered is:

```
1 select last_name,hire_date from employee where department_id=(select department_id from employee where last_name='Zlotkey')
2 and last_name not in('Zlotkey')
```

The results table shows two rows:

LAST_NAME	HIRE_DATE
Davies	05/03/1997
Kohli	04/03/2000

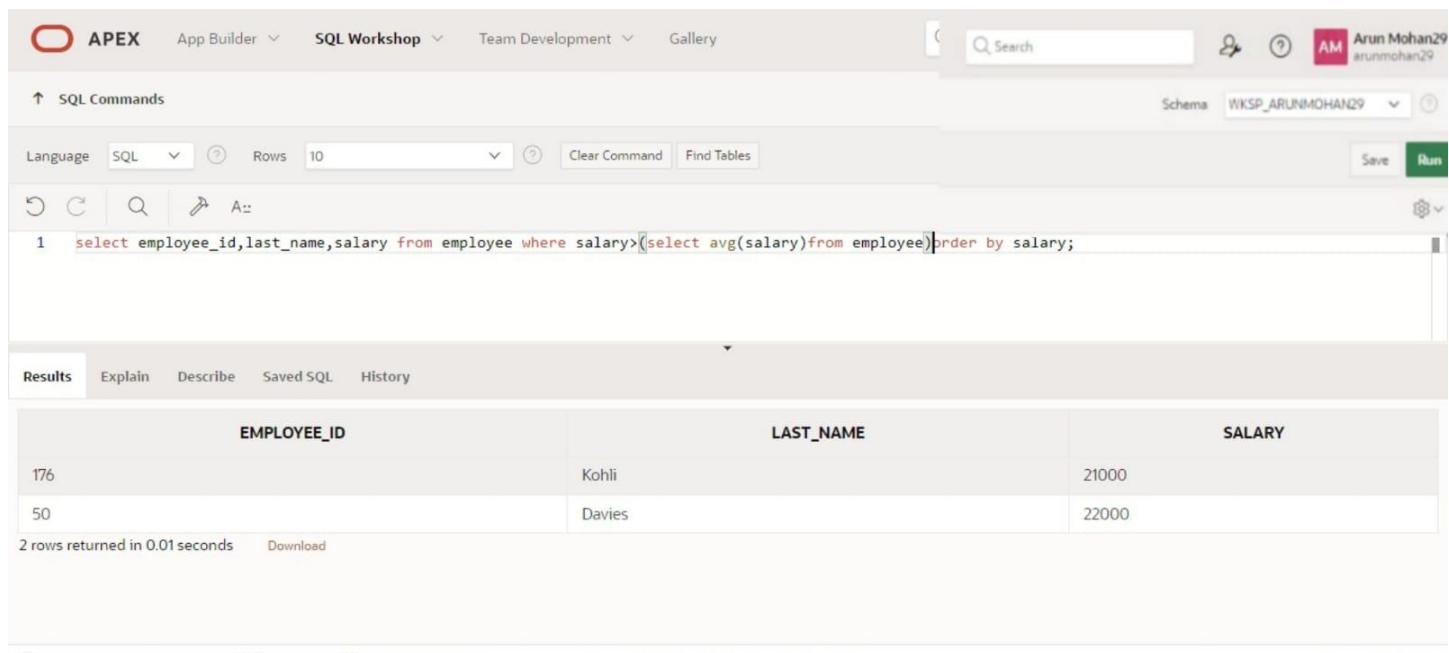
2 rows returned in 0.00 seconds

9. Show the structure of the JOB_GRADES table. Create a query that displays the name, job, department name, salary, and grade for all employees

QUERY:

```
SELECT e.last_name, e.job_id, d.dept_name, e.salary, j.grade_level  
FROM emp18 e JOIN dept18 d  
ON (e.dept_id = d.dept_id)  
JOIN job_grade j  
ON (e.salary BETWEEN j.lowest_sal AND j.highest_sal);
```

OUTPUT:



The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. The right side shows the user profile 'Arun Mohan29' and the schema 'WKSP_ARUNMOHAN29'. The main area is titled 'SQL Commands' with a language dropdown set to 'SQL'. The command entered is:

```
1 select employee_id, last_name, salary from employee where salary > (select avg(salary) from employee) order by salary;
```

The results tab is selected, displaying the following data:

EMPLOYEE_ID	LAST_NAME	SALARY
176	Kohli	21000
50	Davies	22000

Below the table, it says '2 rows returned in 0.01 seconds' and has a 'Download' link.

10.Create a query to display the name and hire date of any employee hired after employee Davies.

QUERY:

```
SELECT e.last_name, e.hire_date FROM emp18 e, emp18 davies
```

```
WHERE davies.last_name = 'Davies'
```

```
AND davies.hire_date < e.hire_date;
```

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. A search bar and user profile for 'Arun Mohan29' are also present. The main workspace displays a SQL command in the editor:

```
1 select employee_id, last_name from employee where department_id=(select department_id from employee where last_name like'%u%');|
```

The results tab is selected, showing the output of the query:

EMPLOYEE_ID	LAST_NAME
4	lucifer

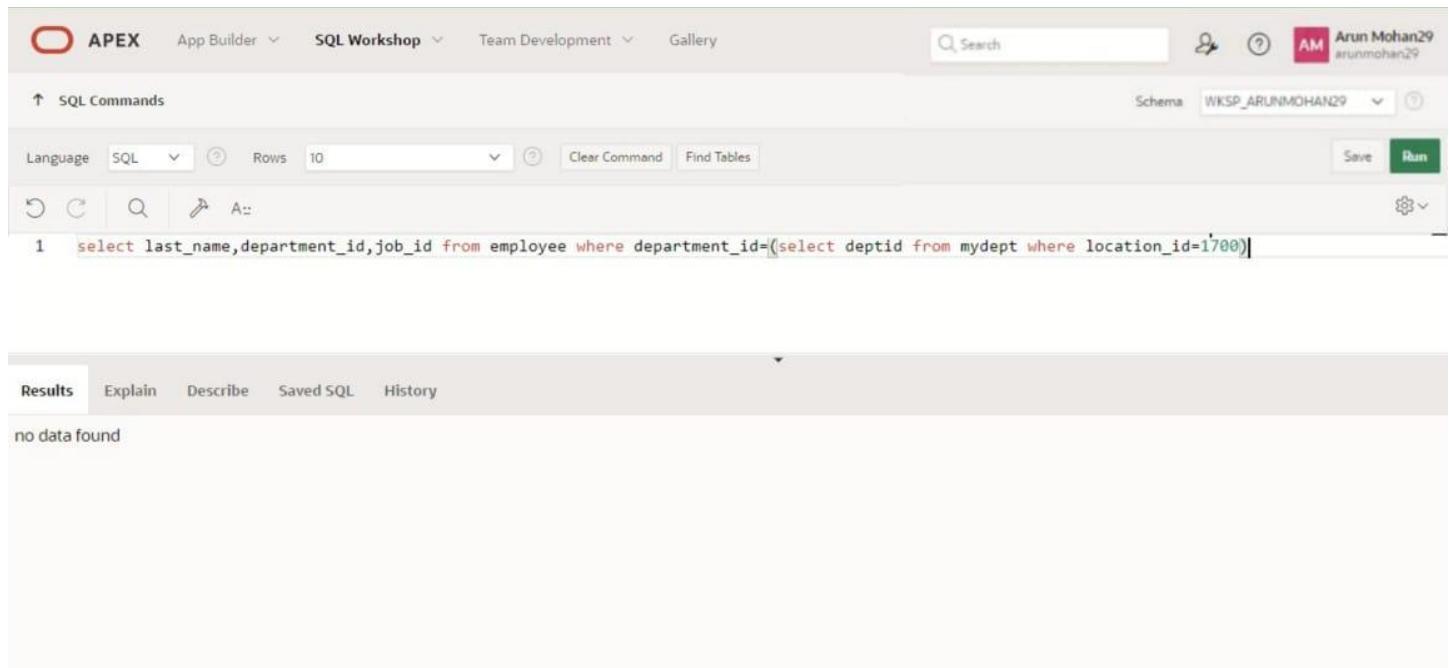
Below the results, it says "1 rows returned in 0.01 seconds". The bottom footer includes copyright information for Oracle and the APEX version.

11. Display the names and hire dates for all employees who were hired before their managers, along with their manager's names and hire dates. Label the columns Employee, Emp Hired, Manager, and Mgr Hired, respectively.

QUERY:

```
SELECT e.last_name AS Employee, e.hire_date AS Emp_Hired,  
e.manager_name AS Manager, m.hire_date AS Mgr_Hired  
FROM emp18 e  
JOIN emp18|m ON e.manager_name = m.last_name  
WHERE e.hire_date < m.hire_date;
```

OUTPUT:



The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. The right side shows a user profile for Arun Mohan29. The main area is a SQL editor with the following content:

```
1 select last_name,department_id,job_id from employee where department_id=(select deptid from mydept where location_id=1700)
```

Below the editor, a results tab is selected, showing the message "no data found".

Evaluation Procedure	Marks awarded
Query(5)	
Execution (5)	
Viva(5)	
Total (15)	
Faculty Signature	

RESULT

AGGREGATING DATA USING GROUP FUNCTIONS

EX_NO : 8

DATE:

1. Group functions work across many rows to produce one result per group.

True/False

TRUE

2. Group functions include nulls in calculations.

True/False

FALSE

3. The WHERE clause restricts rows prior to inclusion in a group calculation.

True/False

FALSE

4. Find the highest, lowest, sum, and average salary of all employees. Label the columns Maximum, Minimum, Sum, and Average, respectively. Round your results to the nearest whole number

QUERY:

```
select Round(Max (salary),0)"Maximum", Round (Min (salary),0) "Minimum",
```

```
round(sum(salary),0)"sum", round (avg(salary),0) "Average" from EMPB;
```

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop, Team Development, and Gallery. A search bar and user profile for 'Arun Mohan29' are also present. The main area is titled 'SQL Commands' with a schema dropdown set to 'WKSP_ARUNMOHAN29'. The SQL editor contains the following query:

```
1 select last_name,salary from employee where manager_id=(select manager_id from employee where manager_name='King');
```

The results tab is selected, displaying the output of the query:

LAST_NAME	SALARY
Davies	22000
lucifer	11000

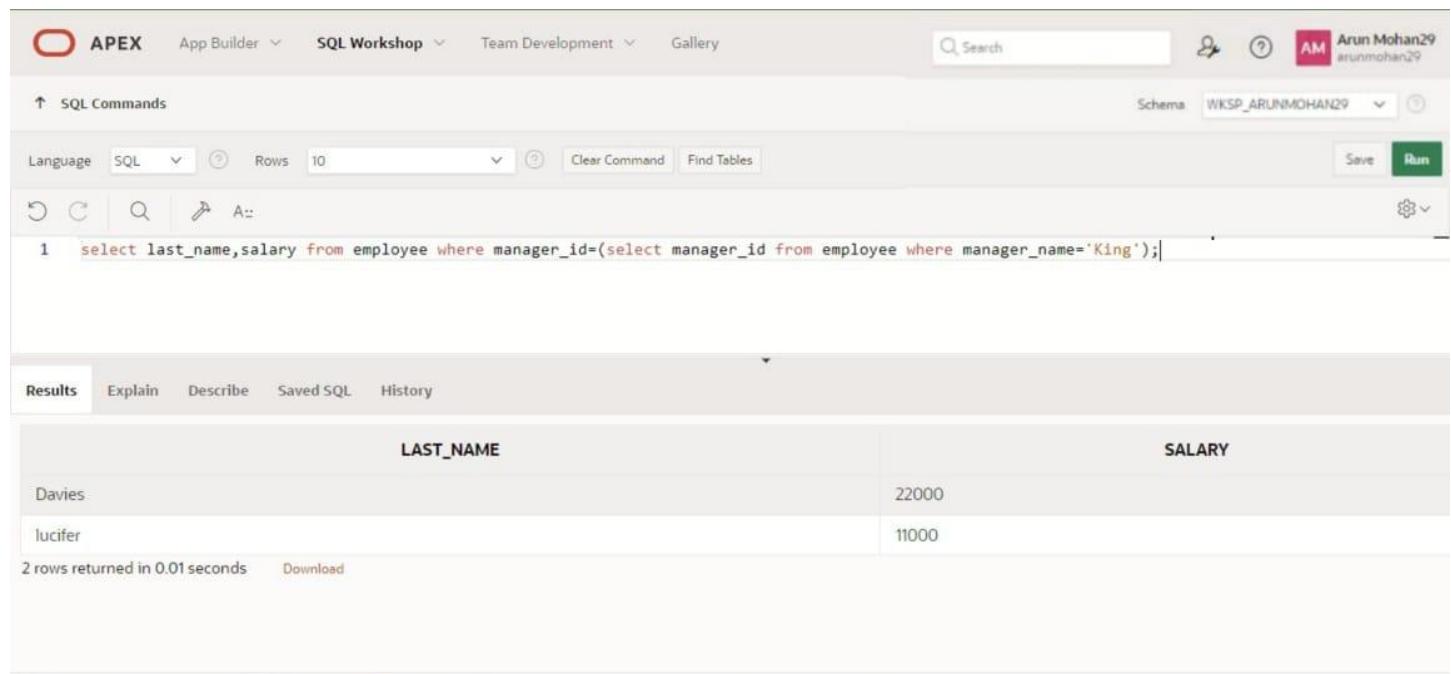
Below the table, it says '2 rows returned in 0.01 seconds' and there is a 'Download' link.

5.Modify the above query to display the minimum, maximum, sum, and average salary for each job type.

QUERY:

```
select job_id ,Round(MAX(salary),0) "MAXIMUM",Round (Min(salary),0)"Minimum",Round  
(SUM(Salary),0)"sum" ,Round (AVg (salary),0)"average" from EMPB group by job_id;
```

OUTPUT:



The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. The right side shows a user profile for Arun Mohan29. The main workspace has tabs for SQL Commands, SQL (selected), and Row limit (10). Below these are buttons for Clear Command, Find Tables, Save, and Run. The SQL command entered is:

```
1 select last_name,salary from employee where manager_id=(select manager_id from employee where manager_name='King'));
```

The Results tab is selected, displaying the output:

LAST_NAME	SALARY
Davies	22000
lucifer	11000

Below the table, it says "2 rows returned in 0.01 seconds" and there is a "Download" link.

6. Write a query to display the number of people with the same job. Generalize the query so that the user in the HR department is prompted for a job title.

QUERY:

```
select job_id, count(*) from EMPB group by job_id ;
```

```
select job_id, count(*) from EMPB where job_id='47' group by job_id ;
```

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. On the right, there's a search bar, a user icon for 'Arun Mohan29', and a schema dropdown set to 'WKSP_ARUNMOHAN29'. The main workspace is titled 'SQL Commands' and contains the following SQL code:

```
1  SELECT deptid FROM mydept MINUS SELECT department_id FROM employee WHERE job_name = 'st_clerk'  
2
```

Below the code, the 'Results' tab is selected, showing the output:

DEPTID
3
78
80

At the bottom left, it says '3 rows returned in 0.01 seconds' and has a 'Download' link. At the bottom right, it shows 'Copyright © 1999, 2023, Oracle and/or its affiliates.' and 'Oracle APEX 23.2.4'.

7.Determine the number of managers without listing them. Label the column Number of Managers. Hint: Use the MANAGER_ID column to determine the number of managers.

QUERY:

```
select count(distinct manager_id )"Number of managers" from empb;
```

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. On the right, there's a search bar, a user profile for 'Arun Mohan29' (arunmohan29), and a schema dropdown set to 'WKSP_ARUNMOHAN29'. The main workspace is titled 'SQL Commands' and contains the following SQL code:

```
1 SELECT deptid FROM mydept MINUS SELECT department_id FROM employee WHERE job_name = 'st_clerk'  
2
```

The results tab is selected, displaying the output:

DEPTID
3
78
80

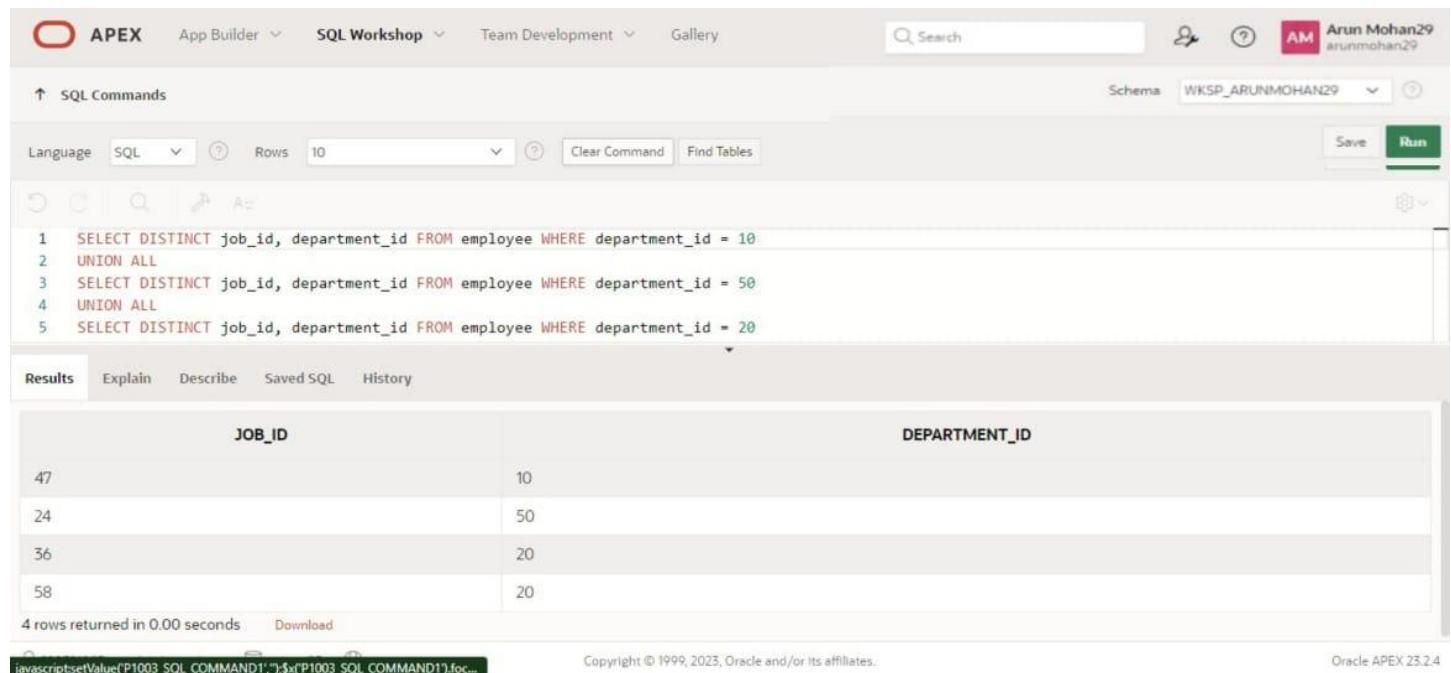
Below the results, it says '3 rows returned in 0.01 seconds' and has a 'Download' link. At the bottom, there's some JavaScript code and copyright information: 'Copyright © 1999, 2023, Oracle and/or its affiliates.' and 'Oracle APEX 23.2.4'.

8.Find the difference between the highest and lowest salaries. Label the column DIFFERENCE

QUERY:

```
select max(salary)-min(salary) difference from empb;
```

OUTPUT:



The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop, Team Development, and Gallery. A search bar and user profile are also present. The main area is titled "SQL Commands" and contains a code editor with the following SQL query:

```
1 SELECT DISTINCT job_id, department_id FROM employee WHERE department_id = 10
2 UNION ALL
3 SELECT DISTINCT job_id, department_id FROM employee WHERE department_id = 50
4 UNION ALL
5 SELECT DISTINCT job_id, department_id FROM employee WHERE department_id = 20
```

Below the code editor, there are tabs for Results, Explain, Describe, Saved SQL, and History. The Results tab is selected, displaying a table with two columns: JOB_ID and DEPARTMENT_ID. The data is as follows:

JOB_ID	DEPARTMENT_ID
47	10
24	50
36	20
58	20

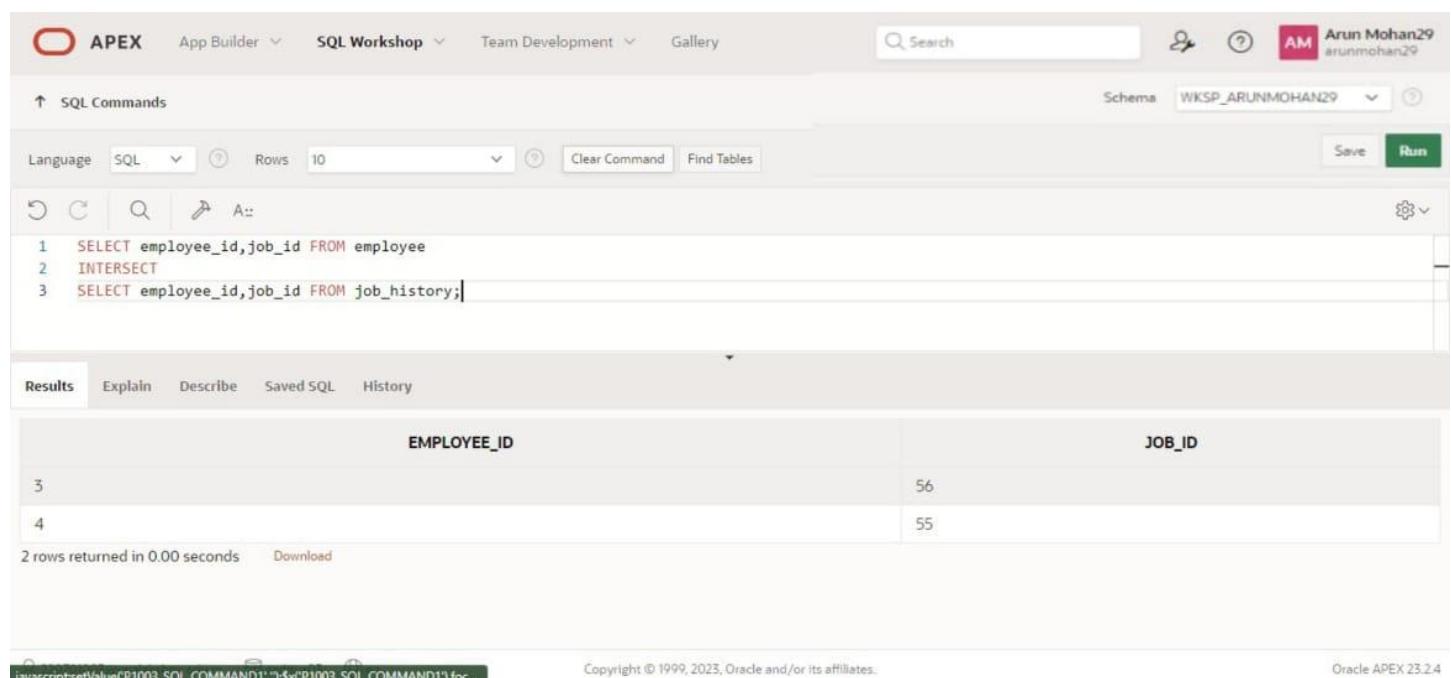
At the bottom left, it says "4 rows returned in 0.00 seconds". At the bottom right, it says "Copyright © 1999, 2023, Oracle and/or Its affiliates." and "Oracle APEX 23.2.4".

9.Create a report to display the manager number and the salary of the lowest-paid employee for that manager. Exclude anyone whose manager is not known. Exclude any groups where the minimum salary is \$6,000 or less. Sort the output in descending order of salary.

QUERY:

```
select manager_id ,MIN(salary) from empb where manager_id is not null group by manager_id having min(salary) >6000 order by min(salary) desc;
```

OUTPUT:



The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. A search bar and user profile are also present. The main workspace has tabs for SQL Commands and Results. The SQL Commands tab contains the following code:

```
1 SELECT employee_id,job_id FROM employee
2 INTERSECT
3 SELECT employee_id,job_id FROM job_history;
```

The Results tab displays the output of the query:

EMPLOYEE_ID	JOB_ID
3	56
4	55

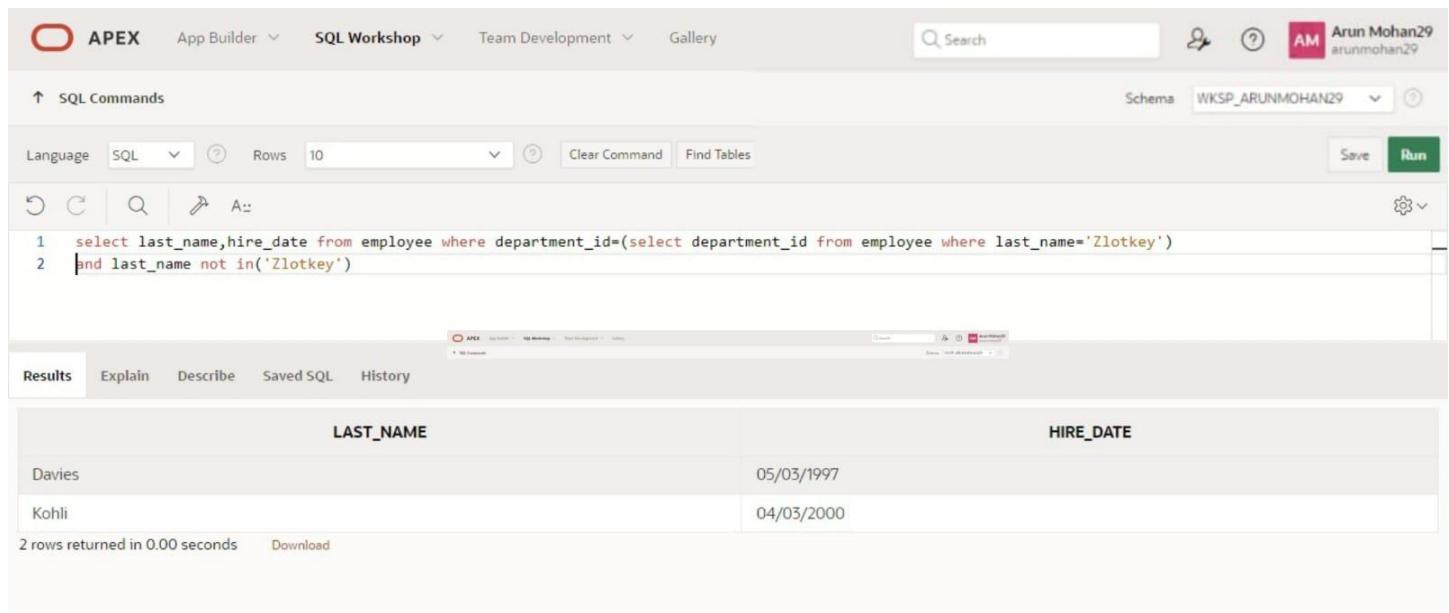
Below the table, it says "2 rows returned in 0.00 seconds". The bottom of the page includes standard footer links and copyright information.

10. Create a query to display the total number of employees and, of that total, the number of employees hired in 1995, 1996, 1997, and 1998. Create appropriate column headings

QUERY:

```
select count(*) total,sum(decode(to_char(hire_date,'YYYY'),1995,1,0)) "1995",sum(decode(to_char(hire_date,'YYYY'),1996,1,0)) "1996",sum(decode(to_char(hire_date,'YYYY'),1997,1,0)) "1997",sum(decode(to_char(hire_date,'YYYY'),1998,1,0)) "1998" from empb;
```

OUTPUT:



The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. The right side shows the user Arun Mohan29 (arunmohan29). The main area has tabs for SQL Commands, SQL (selected), Clear Command, Find Tables, Save, and Run. The SQL editor contains the following code:

```
1 select last_name,hire_date from employee where department_id=(select department_id from employee where last_name='Zlotkey')
2 and last_name not in('Zlotkey')
```

The Results tab is selected, displaying the output:

LAST_NAME	HIRE_DATE
Davies	05/03/1997
Kohli	04/03/2000

Below the table, it says "2 rows returned in 0.00 seconds" and there is a "Download" link.

11.Create a matrix query to display the job, the salary for that job based on department number, and the total salary for that job, for departments 20, 50, 80, and 90, giving each column an appropriate heading

QUERY:

```
select job_id "job", sum(decode(dept_id,20,salary))"Dept20",sum (decode(dept_id ,50, salary)) "dept50",sum (decode(dept_id ,80, salary)) "dept80",sum (decode(dept_id ,90, salary)) "dept90",sum(salary) "TOTAL" from empb group by job_id
```

OUTPUT:

The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. The right side shows the user profile 'Arun Mohan29' and the schema 'WKSP_ARUNMOHAN29'. The main area is titled 'SQL Commands' with tabs for Language (set to SQL), Rows (set to 10), Clear Command, Find Tables, Save, and Run.

The SQL command entered is:

```
1 select employee_id, last_name, salary from employee where salary > (select avg(salary) from employee) order by salary;
```

The results section displays the following data:

EMPLOYEE_ID	LAST_NAME	SALARY
176	Kohli	21000
50	Davies	22000

Below the table, it says '2 rows returned in 0.01 seconds' and has a 'Download' link.

12. Write a query to display each department's name, location, number of employees, and the average salary for all the employees in that department. Label the column name-Location, Number of people, and salary respectively. Round the average salary to two decimal places.

QUERY:

```
select d.dept_name as "dept_name",d.loc as "department location", count(*) "Number of people",round(avg(salary),2) "salary" from dept111 d inner join empb e on(d.dpt_id =e.dept_id ) group by d.dept_name ,d.loc;
```

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. On the right, there is a search bar, user profile for Arun Mohan29, and a schema dropdown set to WKSP_ARUNMOHAN29. The main workspace is titled 'SQL Commands' and contains a SQL editor with the following code:

```
1 select employee_id, last_name from employee where department_id=(select department_id from employee where last_name like'%u%');
```

Below the editor, there are buttons for Language (SQL), Rows (10), Clear Command, Find Tables, Save, and Run. The results tab is selected, showing the output of the query:

EMPLOYEE_ID	LAST_NAME
4	lucifer

1 rows returned in 0.01 seconds [Download](#)

At the bottom of the page, there is a footer with copyright information and a note about Oracle APEX version 23.2.4.

Evaluation Procedure	Marks awarded
Query(5)	
Execution (5)	
Viva(5)	
Total (15)	
Faculty Signature	

RESULT:

SUB QUERIES

EX_NO:9

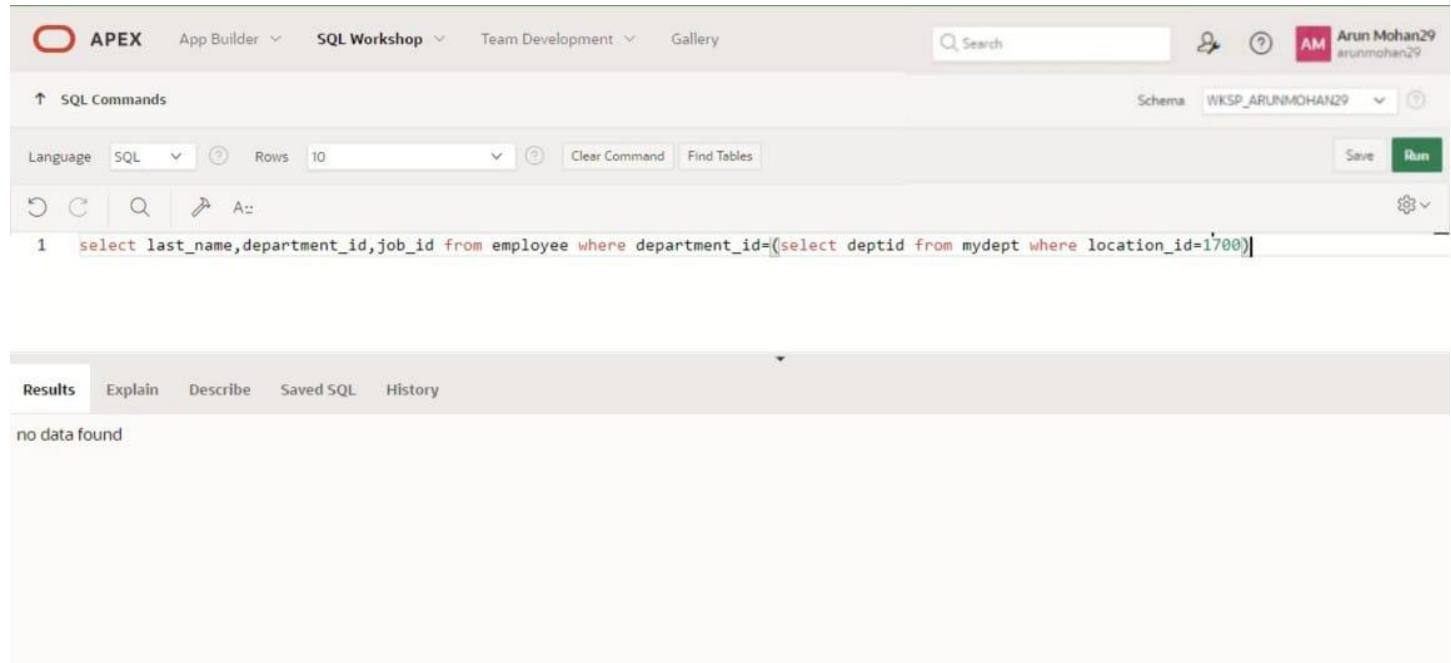
DATE: _____

1.)The HR department needs a query that prompts the user for an employee last name. The query then displays the last name and hire date of any employee in the same department as the employee whose name they supply (excluding that employee). For example, if the user enters Zlotkey, find all employees who work with Zlotkey (excluding Zlotkey).

QUERY:

```
select last_name,hire_date from employees where department_id=(select department_id from employees where last_name='Janu') and last_name not in('Janu');
```

OUTPUT:



The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. On the right, there's a search bar, a user profile for 'Arun Mohan29', and a schema dropdown set to 'WKSP_ARUNMOHAN29'. The main workspace is titled 'SQL Commands' and contains a command line with the following SQL code:

```
1 select last_name,department_id,job_id from employee where department_id=(select deptid from mydept where location_id=1700)
```

Below the command line, there are buttons for Language (SQL), Rows (10), Clear Command, Find Tables, Save, and Run. The results tab is selected, showing the message "no data found".

2.) Create a report that displays the employee number, last name, and salary of all employees who earn more than the average salary. Sort the results in order of ascending salary.

QUERY:

```
select employee_id,last_name,salary from employees where salary>(select avg(salary) from employees) order by salary;
```

OUTPUT:

The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. On the right, there's a search bar, user profile (Arun Mohan29), and a workspace dropdown (WKSP_ARUNMOHAN29). The main area has tabs for SQL Commands, Explain, Describe, Saved SQL, and History. The SQL Commands tab shows the following query:

```
1 select last_name,salary from employee where manager_id=(select manager_id from employee where manager_name='King');
```

The Results tab displays the output of the query:

LAST_NAME	SALARY
Davies	22000
lucifer	11000

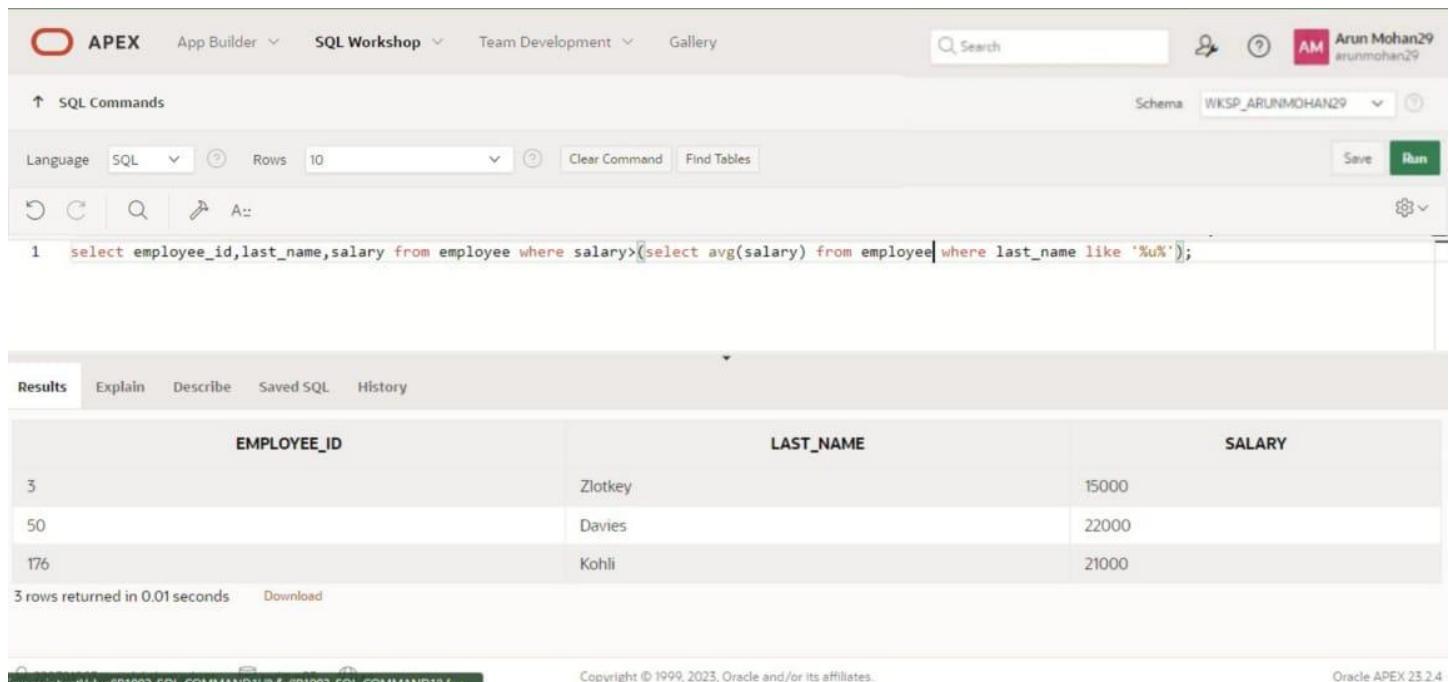
Below the table, it says "2 rows returned in 0.01 seconds" and has a "Download" link.

3.) Write a query that displays the employee number and last name of all employees who work in a department with any employee whose last name contains a u.

QUERY:

```
select employee_id,last_name from employees where department_id=(select department_id from employees where last_name like'%u%');
```

OUTPUT:



The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop, Team Development, and Gallery. The user is signed in as Arun Mohan29. The SQL Commands tab is active, showing the following SQL code:

```
1 select employee_id, last_name, salary from employee where salary > (select avg(salary) from employee where last_name like '%u%');
```

The Results tab is selected, displaying the query results in a grid format:

EMPLOYEE_ID	LAST_NAME	SALARY
3	Zlotkey	15000
50	Davies	22000
176	Kohli	21000

Below the results, it says "3 rows returned in 0.01 seconds". The bottom status bar shows the URL as <https://apex.oracle.com/pls/apex/f?p=100:1:1454581003::NO:::>, the copyright notice "Copyright © 1999, 2023, Oracle and/or its affiliates.", and the version "Oracle APEX 23.2.4".

4.) The HR department needs a report that displays the last name, department number, and job ID of all employees whose department location ID is 1700.

QUERY:

```
select last_name,department_id,job_id from employees where department_id=(select dept_id from departments  
where location_id=1700);
```

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop, Team Development, and Gallery. On the right, there's a search bar, a user profile for 'Arun Mohan29', and session information 'WKSP_ARUNMOHAN29'. The main workspace is titled 'SQL Commands' and contains the following SQL code:

```
1 SELECT deptid FROM mydept MINUS SELECT department_id FROM employee WHERE job_name = 'st_clerk'  
2
```

Below the code, the 'Results' tab is selected, showing the output:

DEPTID
3
78
80

At the bottom left, it says '3 rows returned in 0.01 seconds' and 'Download'. At the bottom right, it says 'Copyright © 1999, 2023, Oracle and/or its affiliates.' and 'Oracle APEX 23.2.4'.

5.)Create a report for HR that displays the last name and salary of every employee who reports to King.

QUERY:

```
select last_name,salary from employees where manager_id=(select manager_id from employees where manager_name='King');
```

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop, Team Development, and Gallery. The SQL Workshop tab is selected. The right side of the header shows the user 'Arun Mohan29' and their schema 'WKSP_ARUNMOHAN29'. The main workspace is titled 'SQL Commands' and contains the following SQL code:

```
1  SELECT deptid FROM mydept MINUS SELECT department_id FROM employee WHERE job_name = 'st_clerk'
2
```

Below the code, there are tabs for Results, Explain, Describe, Saved SQL, and History. The Results tab is selected, displaying the output:

DEPTID
3
78
80

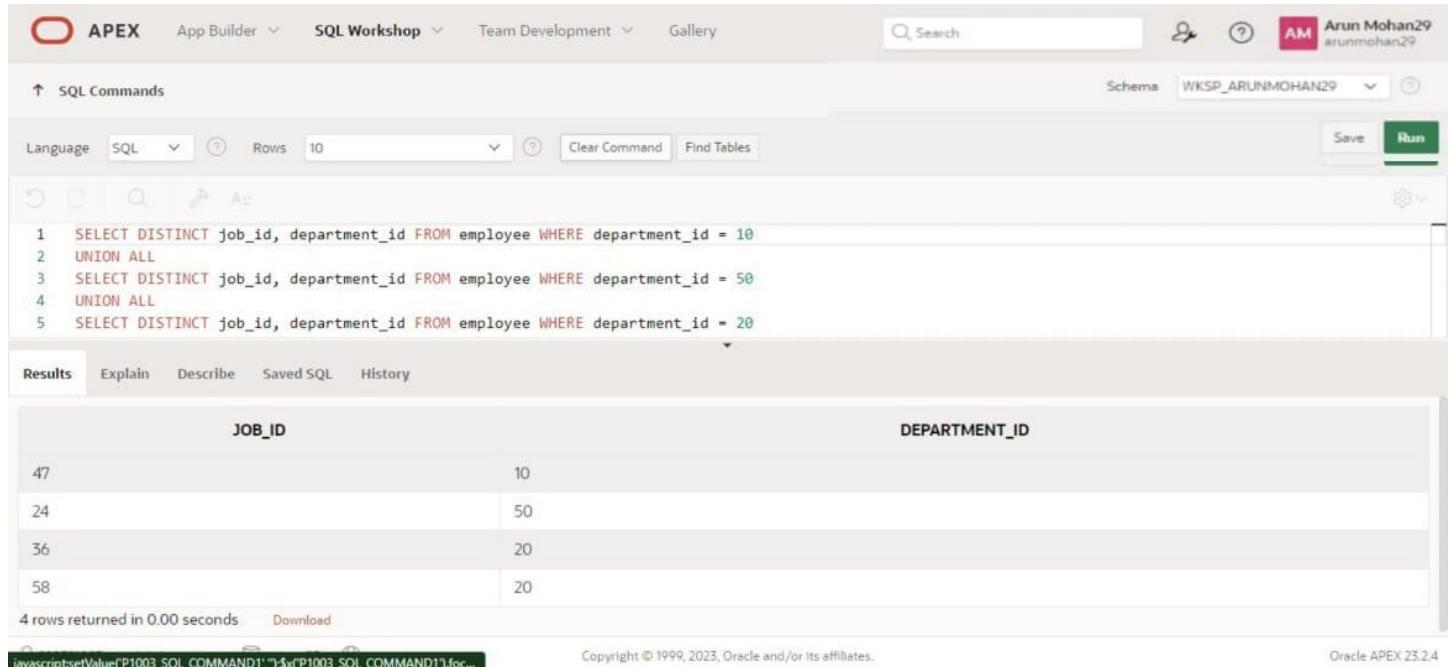
At the bottom left, it says '3 rows returned in 0.01 seconds' and 'Download'. At the bottom right, it says 'Copyright © 1999, 2023, Oracle and/or its affiliates.' and 'Oracle APEX 23.2.4'.

6.) Create a report for HR that displays the department number, last name, and job ID for every employee in the Executive department.

QUERY:

```
select department_id, last_name, job_id from employees where department_id in (select dept_id from departments where dept_name='Executive');
```

OUTPUT:



The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop, Team Development, and Gallery. On the right, there's a search bar, user profile for Arun Mohan29, and a schema dropdown set to WKSP_ARUNMOHAN29. Below the toolbar, the command input area has a language selector set to SQL, row limit set to 10, and buttons for Clear Command, Find Tables, Save, and Run. The main workspace displays the following SQL code:

```
1 SELECT DISTINCT job_id, department_id FROM employee WHERE department_id = 10
2 UNION ALL
3 SELECT DISTINCT job_id, department_id FROM employee WHERE department_id = 50
4 UNION ALL
5 SELECT DISTINCT job_id, department_id FROM employee WHERE department_id = 20
```

The Results tab is selected, showing a table with two columns: JOB_ID and DEPARTMENT_ID. The data returned is:

JOB_ID	DEPARTMENT_ID
47	10
24	50
36	20
58	20

Below the table, it says "4 rows returned in 0.00 seconds". The bottom of the page includes a JavaScript snippet, copyright information (Copyright © 1999, 2023, Oracle and/or its affiliates.), and a note about Oracle APEX 23.2.4.

7.) Modify the query 3 to display the employee number, last name, and salary of all employees who earn more than the average salary and who work in a department with any employee whose last name contains a u.

QUERY:

```
select employee_id, last_name, salary from employees where salary > (select avg(salary) from employees where last_name like '%u%');
```

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. A search bar and user profile are also present. The main area is titled "SQL Commands". The schema dropdown is set to "WKSP_ARUNMOHAN29". The command input field contains the following SQL code:

```
1 SELECT employee_id, job_id FROM employee
2 INTERSECT
3 SELECT employee_id, job_id FROM job_history;
```

The results tab is selected, displaying the output of the query. The table has two columns: "EMPLOYEE_ID" and "JOB_ID". The data is as follows:

EMPLOYEE_ID	JOB_ID
3	56
4	55

Below the table, it says "2 rows returned in 0.00 seconds" and there is a "Download" link. At the bottom of the page, there is some JavaScript code in the footer and copyright information: "Copyright © 1999, 2023, Oracle and/or its affiliates." and "Oracle APEX 23.2.4".

Evaluation Procedure	Marks awarded
Query(5)	
Execution (5)	
Viva(5)	
Total (15)	
Faculty Signature	

RESULT:

USING THE SET OPERATORS

EX_NO:10

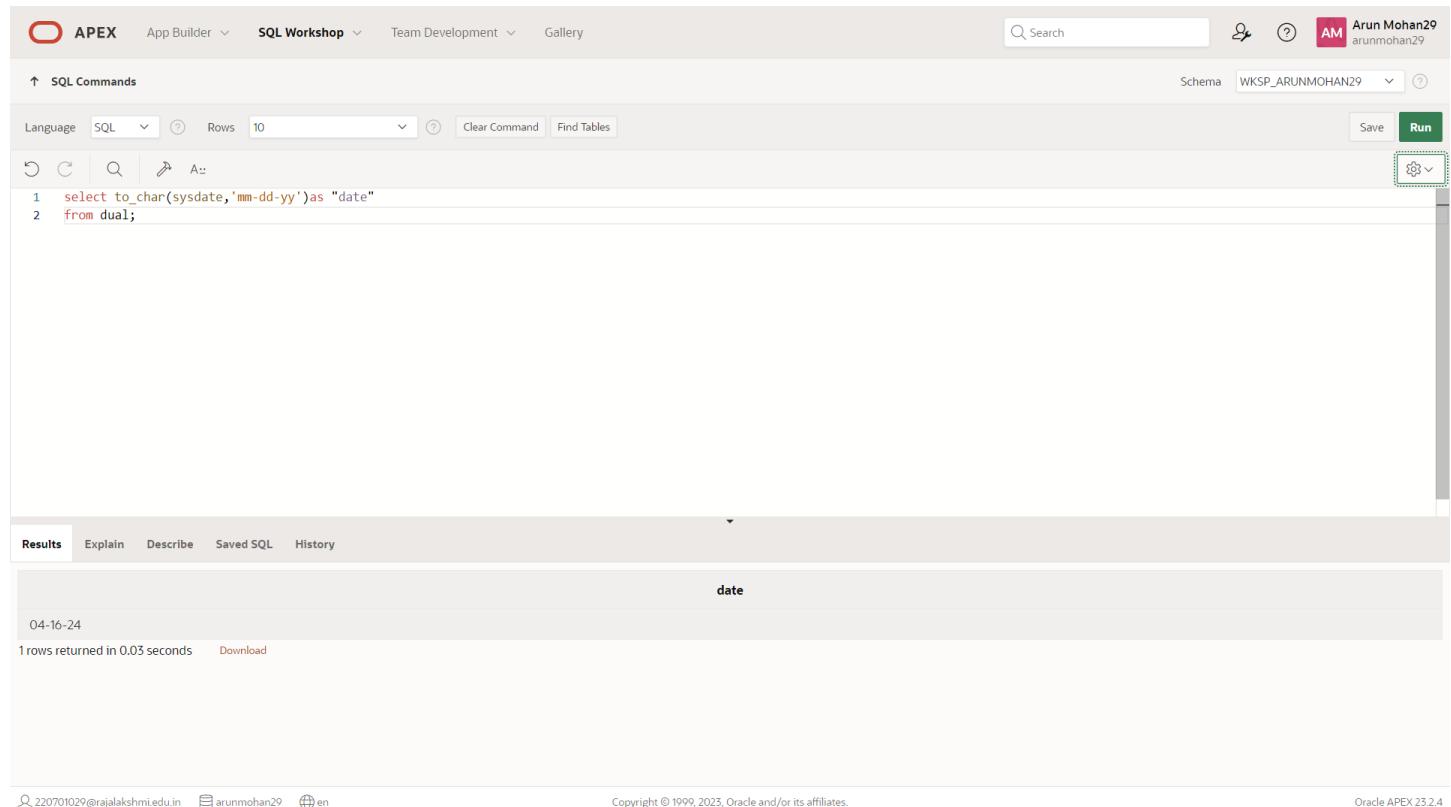
DATE:

- 1.) The HR department needs a list of department IDs for departments that do not contain the job ID ST_CLERK. Use set operators to create this report.

QUERY:

```
select department_id from employees minus select department_id from employees where job_id='st_clerk';
```

OUTPUT:



The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. On the right, there's a user profile for 'Arun Mohan29' and a search bar. The main workspace is titled 'SQL Commands' and contains the following code:

```
1 select to_char(sysdate,'mm-dd-yy')as "date"
2 from dual;
```

Below the code, the 'Results' tab is selected, showing the output:

date
04-16-24

At the bottom, it says '1 rows returned in 0.03 seconds' and provides a 'Download' link. The footer includes copyright information for Oracle and links for user details and help.

2.)The HR department needs a list of countries that have no departments located in them. Display the country ID and the name of the countries. Use set operators to create this report.

QUERY:

```
select country_id,state_province from location minus select country_id,state_province from  
location,departments where location.location_id=departments.location_id;
```

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. On the right, there's a search bar, user information for 'Arun Mohan29', and a 'Run' button. The main workspace has tabs for SQL Commands, Results, Explain, Describe, Saved SQL, and History. The SQL Commands tab contains the following SQL code:

```
1 select employee_id, last_name,salary, round(salary *0.155)as "new salary" from employees
```

The Results tab displays the output of the query:

EMPLOYEE_ID	LAST_NAME	SALARY	new salary
30	Dhoni	20000	3100
35	miller	8999	1395
18	kolhi	14000	2170
176	roger	11000	1705
333	nolan	12000	1860
43	khan	80200	12431

At the bottom, there are footer links for copyright information and Oracle APEX version 23.2.4.

3.) Produce a list of jobs for departments 10, 50, and 20, in that order. Display job ID and department ID using set operators.

QUERY:

```
select job_id,department_id from employees where department_id=10 union
select job_id,department_id from employees where department_id=50 union
select job_id,department_id from employees where department_id=20;
```

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. On the right, there is a search bar, a user icon for 'Arun Mohan29', and a schema dropdown set to 'WKSP_ARUNMOHAN29'. The main workspace is titled 'SQL Commands' and contains the following code:

```
1 update employees set increase = (salary+salary*0.155)-salary
2 select increase from employees;
```

Below the code, the 'Results' tab is selected, showing the output: '8 row(s) updated.' and '0.03 seconds'. The bottom footer includes copyright information for Oracle and links for user profile and help.

4.) Create a report that lists the employee IDs and job IDs of those employees who currently have a job title that is the same as their job title when they were initially hired by the company (that is, they changed jobs but have now gone back to doing their original job).

QUERY:

```
select job_id,employee_id from employees intersect select e.job_id,e.employee_id from employees  
e.job_history j where e.job_id=j.old_job_id;
```

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. The right side shows the user's profile: Arun Mohan29 (arunmohan29). The main workspace is titled "SQL Commands". The language is set to SQL, and the number of rows is set to 10. The command entered is:

```
1 update employees set increase = (salary+salary*0.155)-salary  
2 select increase from employees;
```

The results tab is selected, displaying the output of the query:

INCREASE
3100
1394.845
2170
1705
1860
12451

At the bottom, there are footer links for 220701029@rajalakshmi.edu.in, arunmohan29, and en, along with copyright information: Copyright © 1999, 2023, Oracle and/or its affiliates. and Oracle APEX 23.2.4.

5.)The HR department needs a report with the following specifications: - Last name and department ID of all the employees from the EMPLOYEES table, regardless of whether or not they belong to a department. - Department ID and department name of all the departments from the DEPARTMENTS table, regardless of whether or not they have employees working in them Write a compound query to accomplish this.

QUERY:

```
select first_name||' '||last_name as "Name",department_id from employees union all select dept_name,dept_id  
from departments;
```

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. On the right, there's a search bar, a user icon for 'Arun Mohan29', and session information 'WKSP_ARUNMOHAN29'. The main workspace is titled 'SQL Commands' and contains the following SQL code:

```
1 update employees set increase = (salary+salary*0.155)-salary  
2 select increase from employees;
```

The results tab is selected, showing a single column named 'INCREASE' with the following data:

INCREASE
3100
1394.845
2170
1705
1860
12431

At the bottom of the page, there are footer links for 'Copyright © 1999, 2023, Oracle and/or its affiliates.', 'Oracle APEX 23.2.4', and language selection ('en').

RESULT:

Evaluation Procedure	Marks awarded
Query(5)	
Execution (5)	
Viva(5)	
Total (15)	
Faculty Signature	

CREATING VIEWS

EX_NO:11

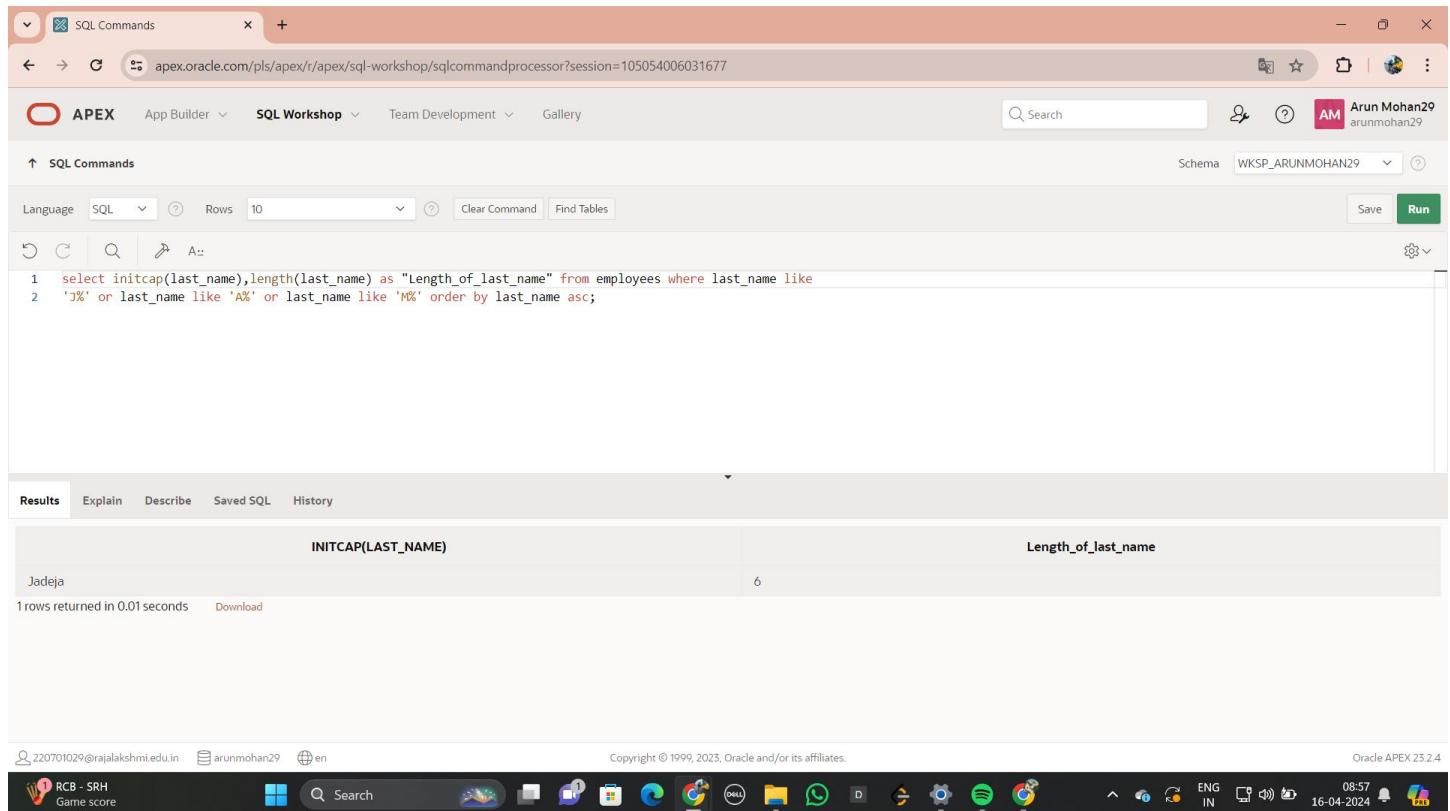
DATE:

1.)Create a view called EMPLOYEE_VU based on the employee numbers, employee names and department numbers from the EMPLOYEES table. Change the heading for the employee name to EMPLOYEE.

QUERY:

```
CREATE OR REPLACE VIEW employees_vu AS SELECT employee_id, last_name employee, department_id
FROM employees;
```

OUTPUT:



The screenshot shows the Oracle APEX SQL Workshop interface. In the SQL Commands tab, a query is being run:

```
1 select initcap(last_name),length(last_name) as "Length_of_last_name" from employees where last_name like
2 'J%' or last_name like 'A%' or last_name like 'M%' order by last_name asc;
```

The results show one row returned:

INITCAP(LAST_NAME)	Length_of_last_name
Jadeja	6

At the bottom, the system status bar shows the user's email (220701029@rajalakshmi.edu.in), session ID (arunmohan29), and the date and time (16-04-2024 08:57 IN).

2.) Display the contents of the EMPLOYEES_VU view.

QUERY:

```
select * from employees_vu;
```

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop (which is selected), Team Development, and Gallery. The right side of the header shows the user's name, Arun Mohan29, and their schema, WKSP_ARUNMOHAN29. The main workspace is titled "SQL Commands". The toolbar below the title bar includes buttons for Language (set to SQL), Rows (set to 10), Clear Command, Find Tables, Save, and Run. The SQL command input field contains the query: "select last_name from employees where last_name like 'Ham%'". Below the input field, the "Results" tab is selected. The results table has a single column labeled "LAST_NAME" and one row labeled "Hammer". A status message at the bottom left indicates "1 rows returned in 0.01 seconds". The bottom of the screen shows the Windows taskbar with various pinned icons and system status information.

LAST_NAME
Hammer

3.)Select the view name and text from the USER_VIEWS data dictionary views

QUERY:

```
SELECT view_name, text FROM user_views;
```

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop (which is selected), Team Development, and Gallery. The right side of the header shows the user's name, Arun Mohan29, and session information. The main area is titled "SQL Commands". The SQL editor contains the following code:

```
1 select last_name,round((sysdate-hire_date)/30,0) as "MONTHS_WORKED" from employees order
2 by round((sysdate-hire_date)/30,0) asc;
```

The "Results" tab is selected, displaying the output of the query:

LAST_NAME	MONTHS_WORKED
kolhi	199
Jadeja	222
Dhoni	239
khan	277
miller	287
pandiya	317

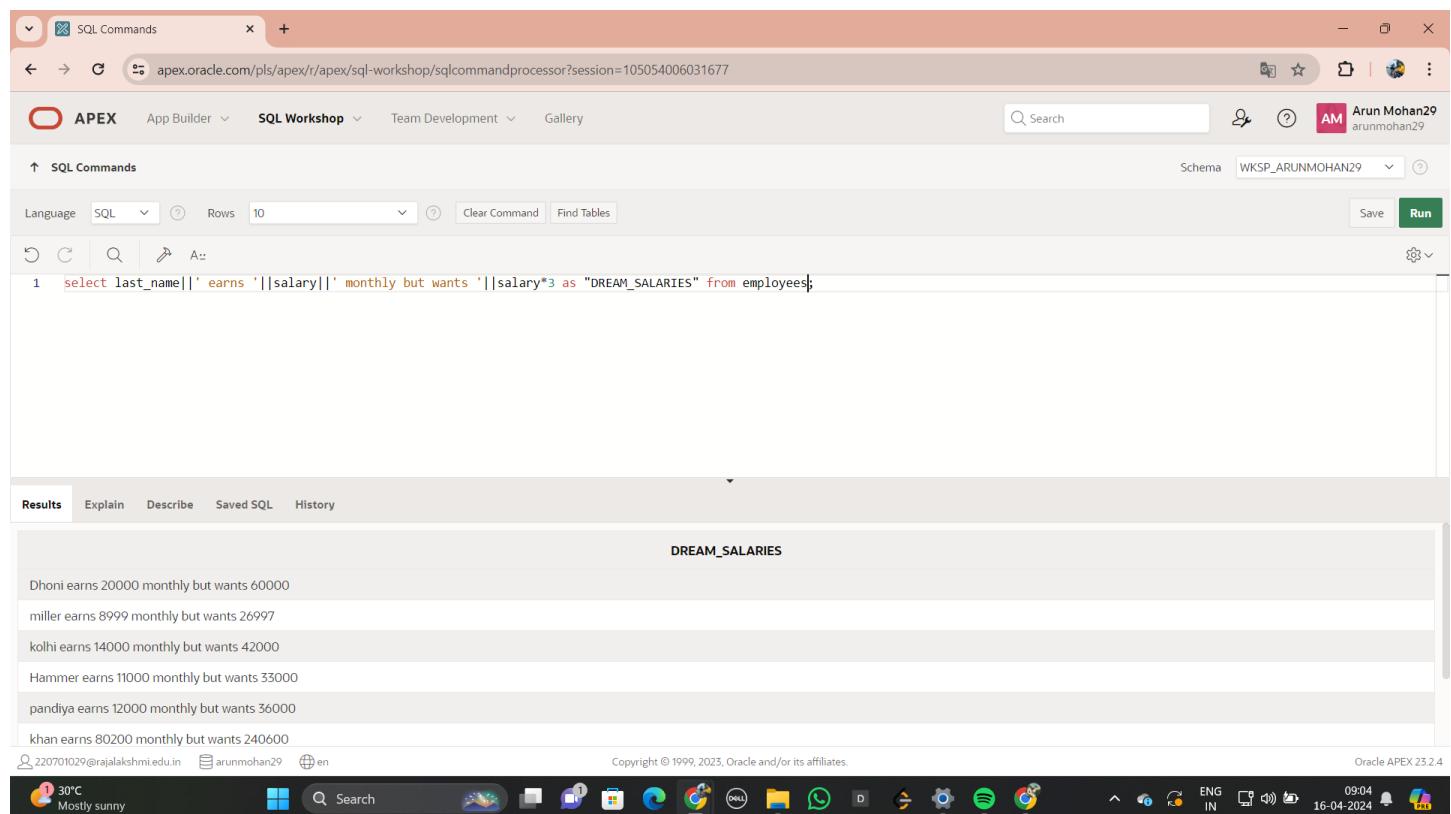
At the bottom, there is a system status bar showing weather (30°C, mostly sunny), search, and various application icons.

4.) Using your EMPLOYEES_VU view, enter a query to display all employees names and department

QUERY:

```
SELECT employee, department_id FROM employees_vu;
```

OUTPUT:



The screenshot shows the Oracle SQL Workshop interface. The query entered is:

```
1 select last_name||' earns'||salary||' monthly but wants '||salary*3 as "DREAM_SALARIES" from employees;
```

The results are displayed in a table with one column labeled "DREAM_SALARIES". The data is:

DREAM_SALARIES
Dhoni earns 20000 monthly but wants 60000
miller earns 8999 monthly but wants 26997
kolhi earns 14000 monthly but wants 42000
Hammer earns 11000 monthly but wants 33000
pandiya earns 12000 monthly but wants 36000
khan earns 80200 monthly but wants 240600

At the bottom, there is a taskbar with various icons and system status information.

5.)Create a view named DEPT50 that contains the employee number, employee last names and department numbers for all employees in department 50.Label the view columns EMPNO, EMPLOYEE and DEPTNO. Do not allow an employee to be reassigned to another department through the view.

QUERY:

```
CREATE VIEW dept50 AS SELECT employee_id empno, last_name employee, department_id deptno FROM employees WHERE department_id = 50 WITH CHECK OPTION CONSTRAINT emp_dept_50;
```

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. On the right, there's a search bar, a user profile for 'Arun Mohan29', and a schema dropdown set to 'WKSP_ARUNMOHAN29'. The main workspace is titled 'SQL Commands' and contains a single line of SQL code:

```
1 select last_name||' earns'||salary||' monthly but wants'||salary*3 as "DREAM_SALARIES" from employees;
```

Below the code, the results tab is selected, showing the output of the query:

DREAM_SALARIES
Dhoni earns 20000 monthly but wants 60000
miller earns 8999 monthly but wants 26997
kolhi earns 14000 monthly but wants 42000
Hammer earns 11000 monthly but wants 33000
pandya earns 12000 monthly but wants 36000
khan earns 80200 monthly but wants 240600

At the bottom, there are footer links for Explain, Describe, Saved SQL, History, and copyright information: Copyright © 1999, 2023, Oracle and/or its affiliates. Oracle APEX 23.2.4.

6.)Display the structure and contents of the DEPT50 view.

QUERY:

Describe dept50;

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. On the right, there's a search bar, a user icon for 'Arun Mohan29', and session information 'WKSP_ARUNMOHAN29'. The main workspace is titled 'SQL Commands' and contains a single line of SQL code: 'select last_name, lpad(salary, 15, '\$') as "SALARY" from employees;'. Below this, the 'Results' tab is selected, displaying the query results in a grid format. The columns are 'LAST_NAME' and 'SALARY'. The data rows are:

LAST_NAME	SALARY
Dhoni	\$\$\$\$\$\$\$\$\$\$20000
miller	\$\$\$\$\$\$\$\$\$\$8999
kolhi	\$\$\$\$\$\$\$\$\$\$14000
Hammer	\$\$\$\$\$\$\$\$\$\$11000
pandiya	\$\$\$\$\$\$\$\$\$\$12000
khan	\$\$\$\$\$\$\$\$\$\$80200

At the bottom of the page, there are footer links for '220701029@rajalakshmi.edu.in', 'arunmohan29', and 'en'. The copyright notice 'Copyright © 1999, 2023, Oracle and/or its affiliates.' and the version 'Oracle APEX 23.2.4' are also present.

7.) Attempt to reassign Matos to department 80

QUERY:

```
UPDATE dept50 SET deptno=80 WHERE employee='Matos';
```

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. On the right, there's a search bar, a user icon for 'Arun Mohan29', and a 'Run' button. The main area is titled 'SQL Commands' with tabs for Language (set to SQL), Rows (set to 10), and buttons for Clear Command and Find Tables. Below this is a toolbar with icons for Undo, Redo, Search, and Run. The SQL editor contains the following code:

```
1  SELECT last_name,hire_date,TO_CHAR(NEXT_DAY(ADD_MONTHS(hire_date, 6),
2  'MONDAY'),'FMDay, "the "FMDD" of "FMMonth, YYYY') AS REVIEW FROM employees;
```

Below the editor is a 'Results' tab, which displays the output of the query:

LAST_NAME	HIRE_DATE	REVIEW
Dhoni	09/07/2004	Monday, the 14 of March, 2005
miller	09/08/2000	Monday, the 12 of March, 2001
kolhi	12/07/2007	Monday, the 09 of June, 2008
Hammer	01/01/1994	Monday, the 04 of July, 1994
pandiya	03/21/1998	Monday, the 28 of September, 1998
khan	07/07/2001	Monday, the 14 of January, 2002

At the bottom, there are footer links for 220701029@rajalakshmi.edu.in, arunmohan29, en, Copyright © 1999, 2023, Oracle and/or its affiliates, and Oracle APEX 23.2.4.

8.) Create a view called SALARY_VU based on the employee last names, department names, salaries, and salary grades for all employees. Use the Employees, DEPARTMENTS and JOB_GRADE tables. Label the column Employee, Department, salary, and Grade respectively.

QUERY:

```
create or replace view salary_vu as select e.last_name "Employee",d.dept_name Department, e.salary  
"Salary",j.grade_level "Grades" from employees e,departments d,job_grade j where e.department_id=d.dept_id  
and e.salary between j.lowest_sal and j.highest_sal;
```

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. On the right, there's a search bar, a user profile for 'Arun Mohan29', and a 'Run' button. The main workspace is titled 'SQL Commands'. It shows the following code:

```
1 SELECT last_name,hire_date,TO_CHAR(hire_date,'Day') as Day from employees|order by  
2 TO_CHAR(hire_date,'Day');
```

Below the code, the 'Results' tab is selected, displaying the output:

LAST_NAME	HIRE_DATE	DAY
miller	09/08/2000	Friday
kolhi	12/07/2007	Friday
Jadeja	01/09/2006	Monday
Hammer	01/01/1994	Saturday
pandiya	03/21/1998	Saturday
khan	07/07/2001	Saturday

At the bottom, there are footer links for copyright information and Oracle APEX version 23.2.4.

Evaluation Procedure	Marks awarded
Query(5)	
Execution (5)	
Viva(5)	
Total (15)	
Faculty Signature	

RESULT:

EXERCISE 12

PRACTICE QUESTIONS

Intro to Constraints; NOT NULL and UNIQUE Constraints

Global Fast Foods has been very successful this past year and has opened several new stores. They need to add a table to their database to store information about each of their store's locations. The owners want to make sure that all entries have an identification number, date opened, address, and city and that no other entry in the table can have the same email address. Based on this information, answer the following questions about the global_locations table. Use the table for your answers.

Global Fast Foods global_locations Table						
NAME	TYPE	LENGTH	PRECISION	SCALE	NULLABLE	DEFAULT
Id						
name						
date_opened						
address						
city						
zip/postal code						
phone						
email						
manager_id						
Emergency contact						

1. What is a “constraint” as it relates to data integrity?

Database can be as reliable as the data in it, and database rules are implemented as Constraint to maintain data integrity.

2. What are the limitations of constraints that may be applied at the column level and at the table level?

- Constraints referring to more than one column are defined at Table Level
- NOT NULL constraint must be defined at column level as per ANSI/ISO SQL standard.

3. Why is it important to give meaningful names to constraints?

- If a constraint is violated in a SQL statement execution, it is easy to identify the cause with user-named constraints.
- It is easy to alter names/drop constraint.

4. Based on the information provided by the owners, choose a datatype for each column. Indicate the length, precision, and scale for each NUMBER datatype.

Global Fast Foods global_locations Table						
NAME	TYPE	DataType	LENGTH	PRECISION	SCALE	NULLABLE
id	pk	NUMBER	6	0		No
name		VARCHAR2	50			
date_opened		DATE				No
address		VARCHAR2	50			No
city		VARCHAR2	30			No
zip_postal_code		VARCHAR2	12			
phone		VARCHAR2	20			
email	uk	VARCHAR2	75			
manager_id		NUMBER	6	0		
emergency_contact		VARCHAR2	20			

5. Use “(nullable)” to indicate those columns that can have null values.

Global Fast Foods global_locations Table

NAME	TYPE	DataType	LENGTH	PRECISION	SCALE	NULLABLE
id	pk	NUMBER	6	0		No
name		VARCHAR2	50			Yes
date_opened		DATE				No
address		VARCHAR2	50			No
city		VARCHAR2	30			No
zip_postal_code		VARCHAR2	12			Yes
phone		VARCHAR2	20			Yes
email	uk	VARCHAR2	75			Yes
manager_id		NUMBER	6	0		Yes
emergency_contact		VARCHAR2	20			Yes

6. Write the CREATE TABLE statement for the Global Fast Foods locations table to define the constraints at the column level.

```
CREATE TABLE f_global_locations
( id NUMBER(6,0) CONSTRAINT f_gln_id_pk PRIMARY KEY ,
  name VARCHAR2(50),
  date_opened DATE CONSTRAINT f_gln_dt_opened_nn NOT NULL ENABLE,
  address VARCHAR2(50) CONSTRAINT f_gln_add_nn NOT NULL ENABLE,
  city VARCHAR2(30) CONSTRAINT f_gln_city_nn NOT NULL ENABLE,
  zip_postal_code VARCHAR2(12),
  phone VARCHAR2(20),
  email VARCHAR2(75) CONSTRAINT f_gln_email_uk UNIQUE,
  manager_id NUMBER(6,0),
  emergency_contact VARCHAR2(20)
);
```

7. Execute the CREATE TABLE statement in Oracle Application Express.

Table Created.

8. Execute a DESCRIBE command to view the Table Summary information.

```
DESCRIBE f_global_locations;
```

9. Rewrite the CREATE TABLE statement for the Global Fast Foods locations table to define the UNIQUE constraints at the table level. Do not execute this statement.

NAME	TYPE	LENGTH	PRECISION	SCALE	NULLABLE	DEFAULT
id	number	4				
loc_name	varchar2	20			X	
	date					
address	varchar2	30				
city	varchar2	20				
zip_postal	varchar2	20			X	
phone	varchar2	15			X	
email	varchar2	80			X	
manager_id	number	4			X	
contact	varchar2	40			X	

```

CREATE TABLE f_global_locations
( id NUMBER(6,0) CONSTRAINT f_gln_id_pk PRIMARY KEY ,
name VARCHAR2(50),
date_opened DATE CONSTRAINT f_gln_dt_opened_nn NOT NULL ENABLE,
address VARCHAR2(50) CONSTRAINT f_gln_add_nn NOT NULL ENABLE,
city VARCHAR2(30) CONSTRAINT f_gln_city_nn NOT NULL ENABLE,
zip_postal_code VARCHAR2(12),
phone VARCHAR2(20),
email VARCHAR2(75) ,
manager_id NUMBER(6,0),
emergency_contact VARCHAR2(20),
CONSTRAINT f_gln_email_uk UNIQUE(email)
);

```

PRIMARY KEY, FOREIGN KEY, and CHECK Constraints

1. What is the purpose of a
 - PRIMARY KEY
 - FOREIGN KEY
 - CHECK CONSTRAINT
- a. PRIMARY KEY**
Uniquely identify each row in table.
- b. FOREIGN KEY**
Referential integrity constraint links back parent table's primary/unique key to child table's column.
- c. CHECK CONSTRAINT**
Explicitly define condition to be met by each row's fields. This condition must be returned as true or unknown.

2. Using the column information for the animals table below, name constraints where applicable at the table level, otherwise name them at the column level. Define the primary key (animal_id). The license_tag_number must be

ANIMAL_ID	NAM E	LICENSE_TAG_NUMBE R	ADMIT_DATE	ADOPTION_ID	VACCINATION_DATE
101	Spot	35540	10-Oct-2004	205	12-Oct-2004

unique. The admit_date and vaccination_date columns cannot contain null values.

animal_id NUMBER(6)	- PRIMARY KEY
name VARCHAR2(25)	
license_tag_number NUMBER(10)	- UNIQUE
admit_date DATE	-NOT NULL
adoption_id NUMBER(5),	
vaccination_date DATE	-NOT NULL

3. Create the animals table. Write the syntax you will use to create the table.

```
CREATE TABLE animals
( animal_id NUMBER(6,0) CONSTRAINT anl_anl_id_pk PRIMARY KEY ,
  name VARCHAR2(25),
  license_tag_number NUMBER(10,0) CONSTRAINT anl_l_tag_num_uk UNIQUE,
  admit_date DATE CONSTRAINT anl_adt_dat_nn NOT NULL ENABLE,
  adoption_id NUMBER(5,0),
  vaccination_date DATE CONSTRAINT anl_vcc_dat_nn NOT NULL ENABLE
);
```

4. Enter one row into the table. Execute a SELECT * statement to verify your input. Refer to the graphic below for input.

```
INSERT INTO animals (animal_id, name, license_tag_number, admit_date, adoption_id, vaccination_date)
VALUES( 101, 'Spot', 35540, TO_DATE('10-Oct-2004', 'DD-Mon-YYYY'), 205, TO_DATE('12-Oct-2004', 'DD-Mon-YYYY'));
```

```
SELECT * FROM animals;
```

5. Write the syntax to create a foreign key (adoption_id) in the animals table that has a corresponding primary-key reference in the adoptions table. Show both the column-level and table-level syntax. Note that because you have not actually created an adoptions table, no adoption_id primary key exists, so the foreign key cannot be added to the animals table.

COLUMN LEVEL STATEMENT:

```
ALTER TABLE animals
MODIFY ( adoption_id NUMBER(5,0) CONSTRAINT anl_adopt_id_fk REFERENCES adoptions(id))
```

ENABLE);

TABLE LEVEL STATEMENT:

```
ALTER TABLE animals ADD CONSTRAINT anl_adopt_id_fk FOREIGN KEY (adoption_id)
REFERENCES adoptions(id) ENABLE;
```

6. What is the effect of setting the foreign key in the ANIMAL table as:

a. ON DELETE CASCADE

```
ALTER TABLE animals
ADD CONSTRAINT anl_adopt_id_fk FOREIGN KEY (adoption_id)
REFERENCES adoptions(id) ON DELETE CASCADE ENABLE ;
```

b. ON DELETE SET NULL

```
ALTER TABLE animals
ADD CONSTRAINT anl_adopt_id_fk FOREIGN KEY (adoption_id)
REFERENCES adoptions(id) ON DELETE SET NULL ENABLE ;
```

7. What are the restrictions on defining a CHECK constraint?

- I cannot specify check constraint for a view however in this case I could use WITH CHECK OPTION clause
- I am restricted to columns from self table and fields in self row.
- I cannot use subqueries and scalar subquery expressions.

Evaluation Procedure	Marks awarded
Query(5)	
Execution (5)	
Viva(5)	
Total (15)	
Faculty Signature	

PRACTICE PROBLEM

Managing Constraints

Using Oracle Application Express, click the SQL Workshop tab in the menu bar. Click the Object Browser and verify that you have a table named copy_d_clients and a table named copy_d_events. If you don't have these tables in your schema, create them before completing the exercises below. Here is how the original tables are related. The d_clients table has a primary key client_number. This has a primary-key constraint and it is referenced in the foreign-key constraint on the d_events table.

NOTE: The practice exercises use the d_clients and d_events tables in the DJs on Demand database. Students will work with copies of these two tables named copy_d_clients and copy_d_events. Make sure they have new copies of the tables (without changes made from previous exercises). Remember, tables copied using a subquery do not have the integrity constraints as established in the original tables. When using the SELECT statement to view the constraint name, the tablename must be all capital letters.

1. What are four functions that an ALTER statement can perform on constraints?

- ADD
- DROP
- ENABLE
- DISABLE

2. Since the tables are copies of the original tables, the integrity rules are not passed onto the new tables; only the column datatype definitions remain. You will need to add a PRIMARY KEY constraint to the copy_d_clients table. Name the primary key copy_d_clients_pk . What is the syntax you used to create the PRIMARY KEY constraint to the copy_d_clients.table?

```
ALTER TABLE copy_d_clients  
ADD CONSTRAINT copy_d_clt_client_number_pk PRIMARY KEY (client_number);
```

3. Create a FOREIGN KEY constraint in the copy_d_events table. Name the foreign key copy_d_events_fk. This key references the copy_d_clients table client_number column. What is the syntax you used to create the FOREIGN KEY constraint in the copy_d_events table?

```
ALTER TABLE copy_d_events  
ADD CONSTRAINT copy_d_eve_client_number_fk FOREIGN KEY (client_number) REFERENCES  
copy_d_clients (client_number) ENABLE;
```

4. Use a SELECT statement to verify the constraint names for each of the tables. Note that the tablename must be capitalized.

```
SELECT constraint_name, constraint_type, table_name  
FROM user_constraints  
WHERE table_name = UPPER('copy_d_events');
```

- a. The constraint name for the primary key in the copy_d_clients table is_____.

COPY_D_CLT_CLIENT_NUMBER_PK

5. Drop the PRIMARY KEY constraint on the copy_d_clients table. Explain your results.

```
ALTER TABLE copy_d_clients
```

```
DROP CONSTRAINT COPY_D_CLT_CLIENT_NUMBER_PK CASCADE ;
```

6. Add the following event to the copy_d_events table. Explain your results.

ID	NAME	EVENT_DATE	DESCRIPTION	COST	VENUE_ID	PACKAGE_CODE	THEME_CODE	CLIENT_NUMBER
140	Cline Bas Mitzvah	15-Jul-2004	Church and Private Home formal	4500	105	87	77	7125

```
INSERT INTO copy_d_events(client_number,id,name,event_date,description,cost,venue_id,package_code,theme_code)
VALUES(7125,140,'Cline Bas Mitzvah',TO_DATE('15-Jul-2004','dd-Mon-yyyy'),'Church and Private Home formal',4500,105,87,77);
```

RESULT: ORA-02291: integrity constraint (HKUMAR.COPY_D_EVE_CLIENT_NUMBER_FK) violated - parent key not found

7. Create an ALTER TABLE query to disable the primary key in the copy_d_clients table. Then add the values from #6 to the copy_d_events table. Explain your results.

```
ALTER TABLE copy_d_clients
DISABLE CONSTRAINT COPY_D_CLT_CLIENT_NUMBER_PK CASCADE;
```

8. Repeat question 6: Insert the new values in the copy_d_events table. Explain your results.

```
INSERT INTO
copy_d_events(client_number,id,name,event_date,description,cost,venue_id,package_code,theme_code)
VALUES(7125,140,'Cline Bas Mitzvah',TO_DATE('15-Jul-2004','dd-Mon-yyyy'),'Church and Private Home formal',4500,105,87,77);
```

1 row(s) inserted.

9. Enable the primary-key constraint in the copy_d_clients table. Explain your results.

```
ALTER TABLE copy_d_clients
ENABLE CONSTRAINT COPY_D_CLT_CLIENT_NUMBER_PK ;
```

10. If you wanted to enable the foreign-key column and reestablish the referential integrity between these two tables, what must be done?

```
DELETE FROM copy_d_events WHERE
client_number NOT IN ( SELECT client_number FROM copy_d_clients);
```

1 row(s) deleted.

```
ALTER TABLE copy_d_events
ENABLE CONSTRAINT COPY_D_EVE_CLIENT_NUMBER_FK;
```

Table altered.

11. Why might you want to disable and then re-enable a constraint?

Generally to make bulk operations fast, where my input data is diligently sanitized and I am sure, it is safe to save some time in this clumsy process.

12. Query the data dictionary for some of the constraints that you have created. How does the data dictionary identify each constraint type?

Queries are same as in point 2,3, 4 above.

- C - Check constraint
Sub-case - if I see SEARCH_CONDITION something like "FIRST_NAME" IS NOT NULL , its a NOT NULL constraint.
- P - Primary key
- R - Referential integrity (fk)
- U - Unique key

Evaluation Procedure	Marks awarded
Query(5)	
Execution (5)	
Viva(5)	
Total (15)	
Faculty Signature	

EXERCISE 13

Creating Views

1. What are three uses for a view from a DBA's perspective?
 - **Restrict access and display selective columns**
 - **Reduce complexity of queries from other internal systems. So, providing a way to view same data in a different manner.**
 - **Let the app code rely on views and allow the internal implementation of tables to be modified later.**
2. Create a simple view called view_d_songs that contains the ID, title and artist from the DJs on Demand table for each "New Age" type code. In the subquery, use the alias "Song Title" for the title column.

```
CREATE VIEW view_d_songs AS
SELECT d_songs.id, d_songs.title "Song Title", d_songs.artist
from d_songs INNER JOIN d_types ON d_songs.type_code = d_types.code
where d_types.description = 'New Age';
```

3. SELECT * FROM view_d_songs. What was returned?

Results	Explain	Describe	Saved SQL	History
ID	Song Title		ARTIST	
47	Hurrah for Today		The Jubilant Trio	
49	Lets Celebrate		The Celebrants	

2 rows returned in 0.00 seconds [Download](#)

4. REPLACE view_d_songs. Add type_code to the column list. Use aliases for all columns. Or use alias after the CREATE statement as shown.

```
CREATE OR REPLACE VIEW view_d_songs AS
SELECT d_songs.id, d_songs.title "Song Title", d_songs.artist, d_songs.type_code
from d_songs INNER JOIN d_types ON d_songs.type_code = d_types.code
where d_types.description = 'New Age';
```

5. Jason Tsang, the disk jockey for DJs on Demand, needs a list of the past events and those planned for the coming months so he can make arrangements for each event's equipment setup. As the company manager, you do not want him to have access to the price that clients paid for their events. Create a view for Jason to use that displays the name of the event, the event date, and the theme description. Use aliases for each column name.

```
CREATE OR REPLACE VIEW view_d_events_pkgs AS
SELECT evt.name "Name of Event", TO_CHAR(evt.event_date, 'dd-Month-yyyy') "Event date", thm.description "Theme
description"
FROM d_events evt INNER JOIN d_themes thm ON evt.theme_code = thm.code
WHERE evt.event_date <= ADD_MONTHS(SYSDATE,1);
```

6. It is company policy that only upper-level management be allowed access to individual employee salaries. The department managers, however, need to know the minimum, maximum, and average salaries, grouped by department. Use the Oracle database to prepare a view that displays the needed information for department managers.

```
CREATE OR REPLACE VIEW view_min_max_avg_dpt_salary ("Department Id", "Department Name",
"Max Salary", "Min Salary", "Average Salary") AS
SELECT dpt.department_id, dpt.department_name, MAX(NVL(emp.salary,0)), MIN(NVL(emp.salary,0)),
ROUND(AVG(NVL(emp.salary,0)),2)
FROM departments dpt LEFT OUTER JOIN employees emp ON dpt.department_id = emp.department_id
GROUP BY (dpt.department_id, dpt.department_name);
```

DML Operations and Views

Use the DESCRIBE statement to verify that you have tables named copy_d_songs, copy_d_events, copy_d_cds, and copy_d_clients in your schema. If you don't, write a query to create a copy of each.

1. Query the data dictionary USER_UPDATABLE_COLUMNS to make sure the columns in the base tables will allow UPDATE, INSERT, or DELETE. All table names in the data dictionary are stored in uppercase.

```
SELECT owner, table_name, column_name, updatable,insertable, deletable  
FROM user_updatable_columns WHERE LOWER(table_name) = 'copy_d_songs';
```

```
SELECT owner, table_name, column_name, updatable,insertable, deletable  
FROM user_updatable_columns WHERE LOWER(table_name) = 'copy_d_events';
```

```
SELECT owner, table_name, column_name, updatable,insertable, deletable  
FROM user_updatable_columns WHERE LOWER(table_name) = 'copy_d_cds';
```

2. Use the CREATE or REPLACE option to create a view of *all* the columns in the copy_d_songs table called view_copy_d_songs.

```
CREATE OR REPLACE VIEW view_copy_d_songs AS  
SELECT *  
FROM copy_d_songs;
```

```
SELECT * FROM view_copy_d_songs;
```

3. Use view_copy_d_songs to INSERT the following data into the underlying copy_d_songs table. Execute a SELECT * from copy_d_songs to verify your DML command. See the graphic.

ID	TITLE	DURATION	ARTIST	TYPE_CODE
88	Mello Jello	2	The What	4

```
INSERT INTO view_copy_d_songs(id,title,duration,artist,type_code)  
VALUES(88,'Mello Jello','2 min','The What',4);
```

4. Create a view based on the DJs on Demand COPY_D_CDS table. Name the view read_copy_d_cds. Select all columns to be included in the view. Add a WHERE clause to restrict the year to 2000. Add the WITH READ ONLY option.

```
CREATE OR REPLACE VIEW read_copy_d_cds AS
SELECT *
FROM copy_d_cds
WHERE year = '2000'
WITH READ ONLY ;

SELECT * FROM read_copy_d_cds;
```

5. Using the read_copy_d_cds view, execute a DELETE FROM read_copy_d_cds WHERE cd_number = 90;

ORA-42399: cannot perform a DML operation on a read-only view

6. Use REPLACE to modify read_copy_d_cds. Replace the READ ONLY option with WITH CHECK OPTION CONSTRAINT ck_read_copy_d_cds. Execute a SELECT * statement to verify that the view exists.

```
CREATE OR REPLACE VIEW read_copy_d_cds AS
SELECT *
FROM copy_d_cds
WHERE year = '2000'
WITH CHECK OPTION CONSTRAINT ck_read_copy_d_cds;
```

7. Use the read_copy_d_cds view to delete any CD of year 2000 from the underlying copy_d_cds.

DELETE FROM read_copy_d_cds WHERE year = '2000';

8. Use the read_copy_d_cds view to delete cd_number 90 from the underlying copy_d_cds table.

DELETE FROM read_copy_d_cds WHERE cd_number = 90;

9. Use the read_copy_d_cds view to delete year 2001 records.

DELETE FROM read_copy_d_cds WHERE year = '2001';

10. Execute a SELECT * statement for the base table copy_d_cds. What rows were deleted?

Only the one in problem 7 above, not the one in 8 and 9

11. What are the restrictions on modifying data through a view?

DELETE,INSERT,MODIFY restricted if it contains:

Group functions
GROUP BY CLAUSE
DISTINCT
pseudocolumn ROWNUM Keyword

12. What is Moore's Law? Do you consider that it will continue to apply indefinitely? Support your opinion with research from the internet.

It roughly predicted that computing power nearly doubles every year. But Moore also said in 2005 that as per nature of exponential functions, this trend may not continue forever.

13. What is the “singularity” in terms of computing?

Singularity is the hypothesis that the invention of artificial superintelligence will abruptly trigger runaway technological growth, resulting in unfathomable changes to human civilization

Managing Views

1. Create a view from the copy_d_songs table called view_copy_d_songs that includes only the title and artist. Execute a SELECT * statement to verify that the view exists.

CREATE OR REPLACE VIEW view_copy_d_songs ASSELECT title, artistFROM copy_d_songs;SELECT * FROM view_copy_d_songs;

2. Issue a DROP view_copy_d_songs. Execute a SELECT * statement to verify that the view has been deleted.

**DROP VIEW view_copy_d_songs;
SELECT * FROM view_copy_d_songs;**

ORA-00942: table or view does not exist

3. Create a query that selects the last name and salary from the Oracle database. Rank the salaries from highest to lowest for the top three employees.

SELECT * FROM(SELECT last_name, salary FROM employees ORDER BY salary DESC)WHERE ROWNUM <= 3;

4. Construct an inline view from the Oracle database that lists the last name, salary, department ID, and maximum salary for each department. Hint: One query will need to calculate maximum salary by department ID.

SELECT empm.last_name, empm.salary, dptmx.department_idFROM(SELECT dpt.department_id, MAX(NVL(emp.salary,0)) max_dpt_salFROM departments dpt LEFT OUTER JOIN employees emp ON dpt.department_id = emp.department_idGROUP BY dpt.department_id) dptmx LEFT OUTER JOIN employees empm ON dptmx.department_id = empm.department_idWHERE NVL(empm.salary,0) = dptmx.max_dpt_sal;

5. Create a query that will return the staff members of Global Fast Foods ranked by salary from lowest to highest.

SELECT ROWNUM, last_name, salaryFROM(SELECT * FROM f_staffs ORDER BY SALARY);

Indexes and Synonyms

1. What is an index and what is it used for?

Definition: These are schema objects which make retrieval of rows from table faster.

Purpose: An index provides direct and fast access to row in table. They provide indexed path to locate data quickly, so hereby reduce necessity of heavy disk input/output operations.

2. What is a ROWID, and how is it used?

Indexes use ROWID's (base 64 string representation of the row address containing block identifier, row location in the block and the database file identifier) which is the fastest way to access any particular row.

3. When will an index be created automatically?

Primary key/unique key use already existing unique index but if index is not present already, it is created while applying unique/primary key constraint.

4. Create a nonunique index (foreign key) for the DJs on Demand column (cd_number) in the D_TRACK_LISTINGS table. Use the Oracle Application Express SQL Workshop Data Browser to confirm that the index was created.

CREATE INDEX d_tlg_cd_number_fk_i on d_track_listings (cd_number);

5. Use the join statement to display the indexes and uniqueness that exist in the data dictionary for the DJs on Demand D_SONGS table.

```
SELECT ucm.index_name, ucm.column_name, ucm.column_position, uix.uniqueness FROM user_indexes uix  
INNER JOIN user_ind_columns ucm ON uix.index_name = ucm.index_name WHERE ucm.table_name =  
'D_SONGS';
```

6. Use a SELECT statement to display the index_name, table_name, and uniqueness from the data dictionary USER_INDEXES for the DJs on Demand D_EVENTS table.

```
SELECT index_name, table_name, uniqueness FROM user_indexes WHERE table_name = 'D_EVENTS';
```

7. Write a query to create a synonym called dj_tracks for the DJs on Demand d_track_listings table.

CREATE SYNONYM dj_tracks FOR d_track_listings;

8. Create a function-based index for the last_name column in DJs on Demand D_PARTNERS table that makes it possible not to have to capitalize the table name for searches. Write a SELECT statement that would use this index.

CREATE INDEX d_ptr_last_name_idx ON d_partners(LOWER(last_name));

9. Create a synonym for the D_TRACK_LISTINGS table. Confirm that it has been created by querying the data dictionary.

CREATE SYNONYM dj_tracks2 FOR d_track_listings;

SELECT * FROM user_synonyms WHERE table_NAME = UPPER('d_track_listings');

10. Drop the synonym that you created in question

DROP SYNONYM dj_tracks2;

Evaluation Procedure	Marks awarded
Query(5)	
Execution (5)	
Viva(5)	
Total (15)	
Faculty Signature	

RESULT:

OTHER DATABASE OBJECTS

EX NO:14

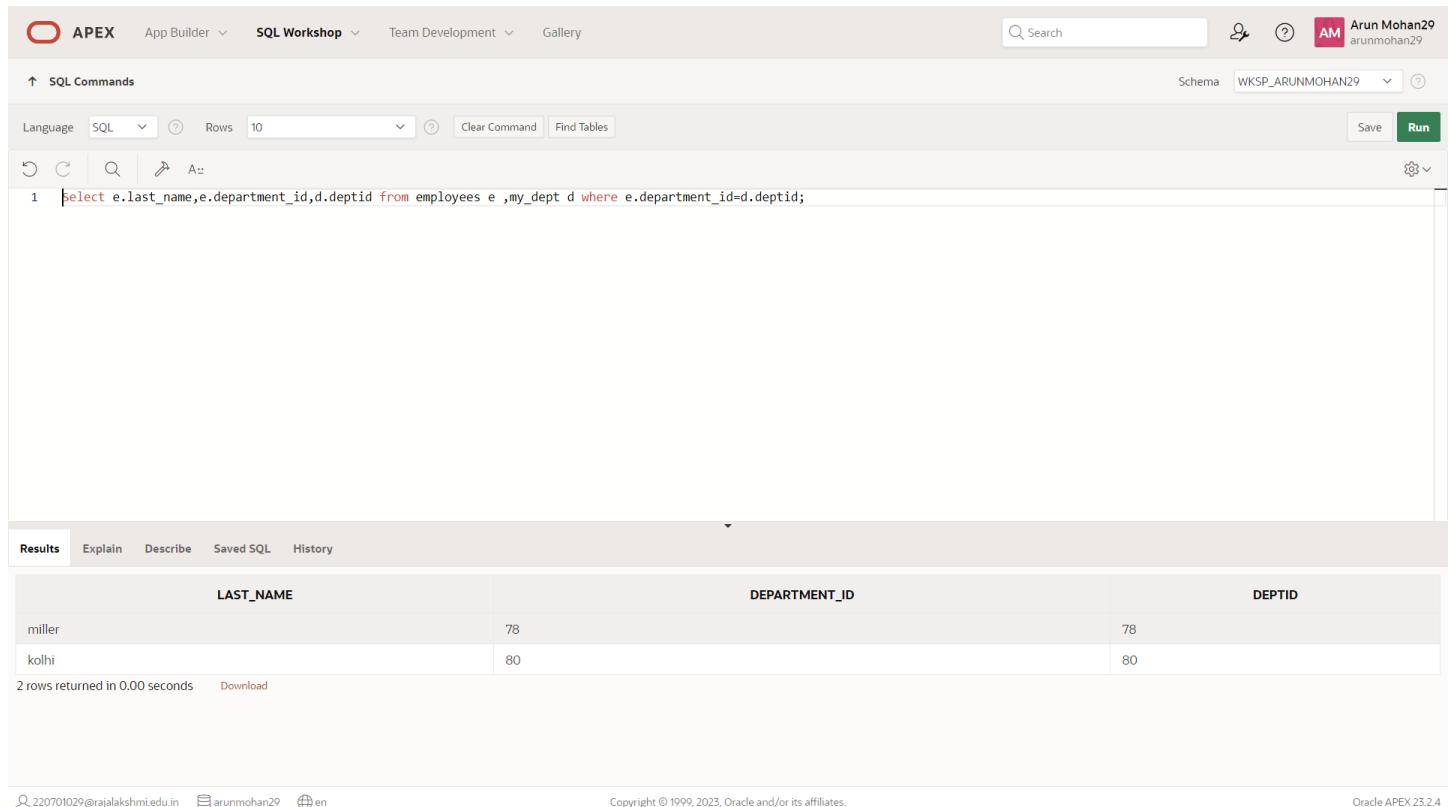
DATE:

1.) Create a sequence to be used with the primary key column of the DEPT table. The sequence should start at 200 and have a maximum value of 1000. Have your sequence increment by ten numbers. Name the sequence DEPT_ID_SEQ

QUERY:

CREATE SEQUENCE dept_id_seq START WITH 200 INCREMENT BY 10 MAXVALUE 1000;

OUTPUT:



The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. On the right, there's a search bar, a user icon for 'Arun Mohan29', and a schema dropdown set to 'WKSP_ARUNMOHAN29'. The main workspace is titled 'SQL Commands' and contains a single line of SQL code: 'select e.last_name,e.department_id,d.deptid from employees e ,my_dept d where e.department_id=d.deptid;'. Below the code, the results tab is selected, displaying a table with three columns: LAST_NAME, DEPARTMENT_ID, and DEPTID. The data shows two rows: one for 'miller' with DEPARTMENT_ID 78 and DEPTID 78, and another for 'kolhi' with DEPARTMENT_ID 80 and DEPTID 80. At the bottom left, it says '2 rows returned in 0.00 seconds'. The bottom footer includes copyright information: 'Copyright © 1999, 2023, Oracle and/or its affiliates.' and 'Oracle APEX 23.2.4'.

LAST_NAME	DEPARTMENT_ID	DEPTID
miller	78	78
kolhi	80	80

2.) Write a query in a script to display the following information about your sequences: sequence name, maximum value, increment size, and last number

QUERY:

```
SELECT sequence_name, max_value, increment_by, last_number FROM user_sequences;
```

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop, Team Development, and Gallery. On the right, there's a search bar, a help icon, and a user profile for 'Arun Mohan29'. The main workspace is titled 'SQL Commands' and contains a single line of SQL code: 'Select last_name,job_id,hire_date from employees where hire_date between '02/20/1998' and '05/01/1998';'. Below the code, the results tab is selected, displaying a table with three columns: LAST_NAME, JOB_ID, and HIRE_DATE. The result for 'nolan' is shown: LAST_NAME is 'nolan', JOB_ID is '244', and HIRE_DATE is '03/21/1998'. At the bottom left, it says '1 rows returned in 0.01 seconds' and has a 'Download' link. The bottom right corner indicates the version 'Oracle APEX 23.2.4'.

LAST_NAME	JOB_ID	HIRE_DATE
nolan	244	03/21/1998

3.) Write a script to insert two rows into the DEPT table. Name your script lab12_3.sql. Be sure to use the sequence that you created for the ID column. Add two departments named Education and Administration. Confirm your additions. Run the commands in your script.

QUERY:

```
INSERT INTO dept VALUES (dept_id_seq.nextval, 'Education');
INSERT INTO dept VALUES (dept_id_seq.nextval, 'Administration');
```

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. On the right, there's a search bar, a user icon for 'Arun Mohan29', and a schema dropdown set to 'WKSP_ARUNMOHAN29'. The main workspace is titled 'SQL Commands' and contains a single line of SQL code:

```
1 select last_name,department_id from employees where department_id in(20,50) order by last_name;
```

Below the code, the results tab is selected, showing the output of the query:

LAST_NAME	DEPARTMENT_ID
nolan	50

At the bottom of the results pane, it says '1 rows returned in 0.00 seconds' and has a 'Download' link. The footer of the page includes links for 220701029@rajalakshmi.edu.in, arunmohan29, en, Copyright © 1999, 2023, Oracle and/or its affiliates, and Oracle APEX 23.2.4.

4.)Create a nonunique index on the foreign key column (DEPT_ID) in the EMP table.

QUERY:

```
CREATE INDEX emp_dept_id_idx ON EMPLOYEES (department_id);
```

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. On the right, there's a search bar, a user icon for 'Arun Mohan29', and a schema dropdown set to 'WKSP_ARUNMOHAN29'. The main workspace is titled 'SQL Commands' and contains a code editor with the following SQL query:

```
1 select last_name as "EMPLOYEE", salary as "MONTHLY SALARY" from employees where (salary between 5000
2 and 12000) and (department_id in(20,50)) order by last_name asc;
3 |
```

Below the code editor, the results tab is selected, showing a single row of data:

EMPLOYEE	MONTHLY SALARY
nolan	12000

At the bottom of the results pane, it says '1 rows returned in 0.01 seconds' and has a 'Download' link. The footer of the page includes user information (email: 220701029@rajalakshmi.edu.in, profile: arunmohan29, language: en), copyright notice (Copyright © 1999, 2023, Oracle and/or its affiliates.), and the software version (Oracle APEX 23.2.4).

5.)Display the indexes and uniqueness that exist in the data dictionary for the EMP table.

QUERY:

```
SELECT index_name,table_name,uniqueness FROM user_indexes WHERE table_name='EMPLOYEES';
```

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. On the right, there's a search bar, a user icon for 'Arun Mohan29' (arunmohan29), and a schema dropdown set to 'WKSP_ARUNMOHAN29'. The main workspace is titled 'SQL Commands' and contains the following SQL code:

```
1 select last_name,hire_date from employees where hire_date like '%1994';
2
```

The results tab is selected, displaying the output of the query:

LAST_NAME	HIRE_DATE
roger	01/01/1994
kumar	02/05/1994

Below the table, it says '2 rows returned in 0.00 seconds' and has a 'Download' link. At the bottom of the page, there are footer links for '220701029@rajalakshmi.edu.in', 'arunmohan29', and 'en'. The copyright notice 'Copyright © 1999, 2023, Oracle and/or its affiliates.' and the version 'Oracle APEX 23.2.4' are also present.

Evaluation Procedure	Marks awarded
Query(5)	
Execution (5)	
Viva(5)	
Total (15)	
Faculty Signature	

RESULT:

CONTROLLING USER ACCESS

EX_NO:15

DATE:

1. What privilege should a user be given to log on to the Oracle Server? Is this a system or an object privilege?

The CREATE SESSION system privilege

2. What privilege should a user be given to create tables?

The CREATE TABLE privilege

3. If you create a table, who can pass along privileges to other users on your table?

You can, or anyone you have given those privileges to by using the WITH GRANT OPTION.

4. You are the DBA. You are creating many users who require the same system privileges. What should you use to make your job easier?

Create a role containing the system privileges and grant the role to the users

5. What command do you use to change your password?

The ALTER USER statement

6. Grant another user access to your DEPARTMENTS table. Have the user grant you query access to his or her DEPARTMENTS table.

Team 2 executes the GRANT statement. GRANT select ON departments TO <user1>;

Team 1 executes the GRANT statement. GRANT select ON departments TO <user2>;

7. Query all the rows in your DEPARTMENTS table.

SELECT * FROM departments;

8. Add a new row to your DEPARTMENTS table. Team 1 should add Education as department number 500. Team 2 should add Human Resources department number 510. Query the other team's table.

Team 1 executes this INSERT statement. INSERT INTO departments(department_id, department_name) VALUES (500, 'Education'); COMMIT;

Team 2 executes this INSERT statement. INSERT INTO departments(department_id, department_name) VALUES (510, 'Administration'); COMMIT;

9. Query the USER_TABLES data dictionary to see information about the tables that you own.
SELECT table_name FROM user_tables;

10. Revoke the SELECT privilege on your table from the other team.

Team 1 revokes the privilege.

```
REVOKE select  
ON departments  
FROM user2;
```

Team 2 revokes the privilege.

```
REVOKE select  
ON departments  
FROM user1;
```

11. Remove the row you inserted into the DEPARTMENTS table in step 8 and save the changes.

Team 1 executes this INSERT statement.

```
DELETE FROM departments  
WHERE department_id = 500;  
COMMIT;
```

Team 2 executes this INSERT statement.

```
DELETE FROM departments  
WHERE department_id = 510;  
COMMIT;
```

Evaluation Procedure	Marks awarded
Query(5)	
Execution (5)	
Viva(5)	
Total (15)	
Faculty Signature	

RESULT:

PROCEDURES AND FUNCTIONS

EX_NO: 17

DATE:

1.) Factorial of a number using function.

QUERY:

```
DECLARE
    fac NUMBER := 1;
    n NUMBER := :1;
BEGIN
    WHILE n > 0 LOOP
        fac := n * fac;
        n := n - 1;
    END LOOP;
    DBMS_OUTPUT.PUT_LINE(fac);
END;
```

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. In the SQL Commands tab, a PL/SQL block is written to calculate the factorial of a number. The code uses a loop to multiply the current value of 'fac' by 'n' until 'n' reaches 0. It then outputs the result using DBMS_OUTPUT.PUT_LINE. The code is as follows:

```
9      END LOOP;
10     result := factorial_result;
11 END;
12 /
13 DECLARE
14     v_result NUMBER;
15 BEGIN
16     factorial(5, v_result);
17
18     DBMS_OUTPUT.PUT_LINE('Factorial: ' || v_result);
19
20 END;
21 /
22 /
23
```

After running the code, the Results tab displays the output: "Factorial: 120". The status bar at the bottom indicates the session ID "220701029@rajalakshmi.edu.in", the user "arunmohan29", and the environment "en". The system also shows the date and time "21-05-2024 08:37" and various system icons.

2.) Write a PL/SQL program using Procedures IN,INOUT,OUT parameters to retrieve the corresponding book information in library.

QUERY:

```
CREATE OR REPLACE PROCEDURE get_book_info (
    p_book_id IN NUMBER,
    p_title IN OUT VARCHAR2,
    p_author OUT VARCHAR2,
    p_year_published OUT NUMBER
)
AS
BEGIN
    SELECT title, author, year_published INTO p_title, p_author, p_year_published
    FROM books
    WHERE book_id = p_book_id;

    p_title := p_title || ' - Retrieved';
EXCEPTION
    WHEN NO_DATA_FOUND THEN
        p_title := NULL;
        p_author := NULL;
        p_year_published := NULL;
END;

DECLARE
    v_book_id NUMBER := 1;
    v_title VARCHAR2(100);
    v_author VARCHAR2(100);
    v_year_published NUMBER;
BEGIN
    v_title := 'Initial Title';

    get_book_info(p_book_id => v_book_id, p_title => v_title, p_author => v_author,
    p_year_published => v_year_published);

    DBMS_OUTPUT.PUT_LINE('Title: ' || v_title);
    DBMS_OUTPUT.PUT_LINE('Author: ' || v_author);
    DBMS_OUTPUT.PUT_LINE('Year Published: ' || v_year_published);
END;
```

OUTPUT:

The screenshot shows the Oracle APEX SQL Workshop interface. In the top navigation bar, the user is in the 'SQL Commands' tab under the 'APEX' section. The URL in the address bar is apex.oracle.com/pls/apex/r/apex/sql-workshop/sqlcommandprocessor?session=4727151959605. The schema selected is 'WKSP_ARUNMOHAN29'. The code editor contains a PL/SQL block:

```
45    v_book_id NUMBER;
46
47    BEGIN
48        -- Call the procedure
49        get_book_info(v_book_id, v_title, v_author, v_year_published);
50
51        -- Print the results
52        DBMS_OUTPUT.PUT_LINE('Book ID: ' || v_book_id);
53        DBMS_OUTPUT.PUT_LINE('Title: ' || v_title);
54        DBMS_OUTPUT.PUT_LINE('Author: ' || v_author);
55        DBMS_OUTPUT.PUT_LINE('Year Published: ' || v_year_published);
56
57    END;
58
59
```

The results tab shows the output of the procedure:

```
Book ID: 2
Title: 1984
Author: George Orwell
Year Published: 1949

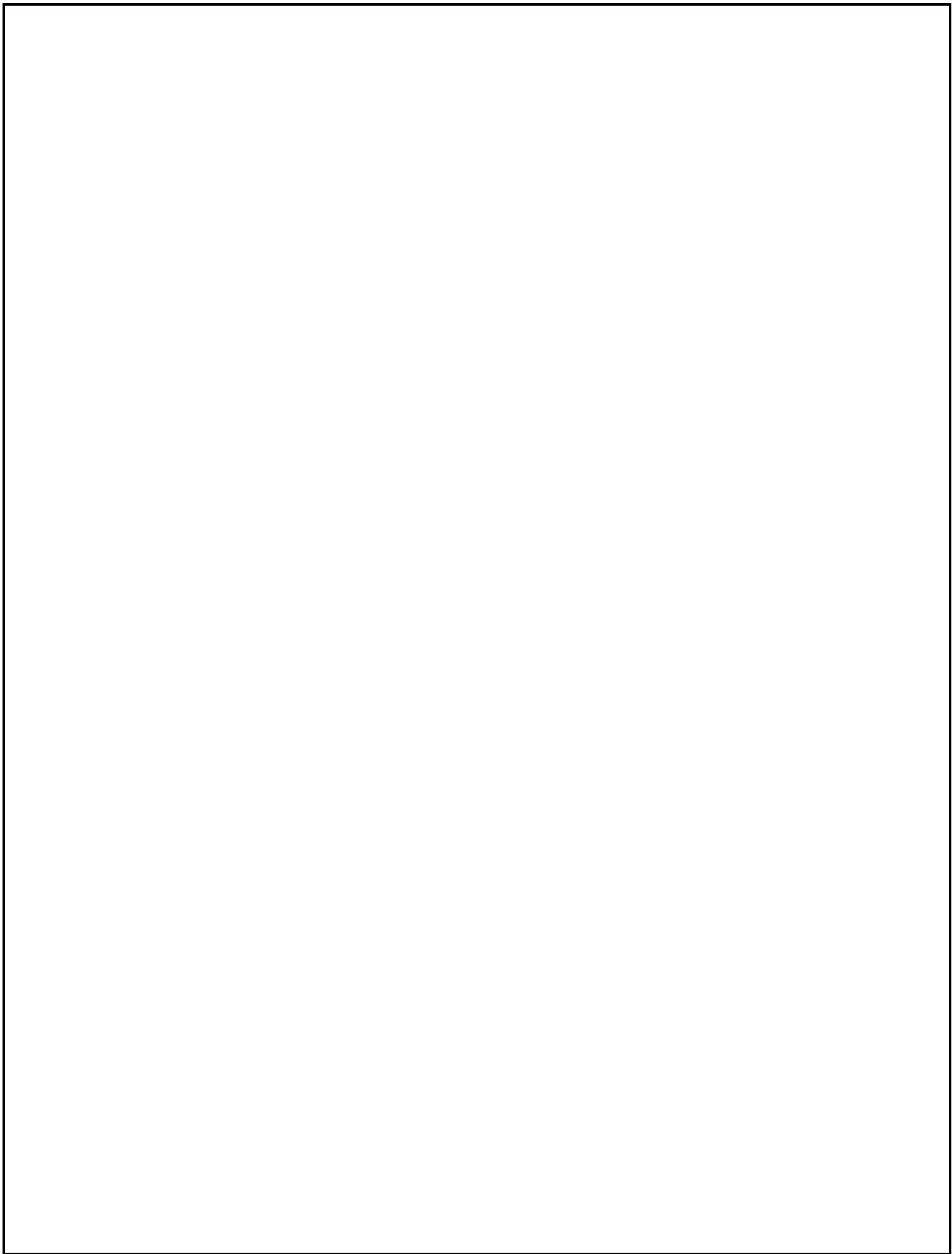
Statement processed.

0.02 seconds
```

The bottom status bar shows system information: 30°C, Partly sunny, Search bar, taskbar icons (File, Help, etc.), and system notifications.

Evaluation Procedure	Marks awarded
Query(5)	
Execution (5)	
Viva(5)	
Total (15)	
Faculty Signature	

RESULT:



MONGO DB

EX_NO: 19

DATE:

1.)Write a MongoDB query to find the restaurant Id, name, borough and cuisine for those restaurants which prepared dish except 'American' and 'Chinees' or restaurant's name begins with letter 'Wil'.

QUERY:

```
db.restaurants.find( { $or: [ { name: /^Wil/ }, { cuisine: { $nin: ['American', 'Chinese'] } } ] , { restaurant_id: 1, name: 1, borough: 1, cuisine: 1 } } );
```

OUTPUT:

The screenshot shows a MongoDB query editor interface. On the left, there is a text input field labeled "Enter a title...". Below it are two dropdown menus: one for "MongoDB" and another with a question mark icon. To the right is a "Run" button with a play icon. The main area contains the MongoDB query:1: { \$or: [{ name: /^Wil/ }, { cuisine: { \$nin: ['American', 'Chinese'] } }] , { restaurant_id: 1, name: 1, borough: 1, cuisine: 1 } }); Output

On the right, the "Output" panel displays the command prompt and the response:mycompiler_mongodb>
mycompiler_mongodb>
[Execution complete with exit code 0]

2.)Write a MongoDB query to find the restaurant Id, name, and grades for those restaurants which achieved a grade of "A" and scored 11 on an ISODate "2014-08-11T00:00:00Z" among many of survey dates.

QUERY:

```
db.restaurants.find( { grades: { $elemMatch: { grade: "A", score: 11, date: ISODate("2014-08-11T00:00:00Z") } } }, { restaurant_id: 1, name: 1, grades: 1 } );
```

OUTPUT:

myCompiler

English Recent

Enter a title...

MongoDB

```
1 { $elemMatch: { grade: "A", score: 11, date: ISODate("2014-08-11T00:00:00Z") } }, { restaurant_id: 1, name: 1, grades: 1 }; Output
```

```
mycompiler_mongodb>
mycompiler_mongodb>

[Execution complete with exit code 0]
```

3.) Write a MongoDB query to find the restaurant Id, name and grades for those restaurants where the 2nd element of grades array contains a grade of "A" and score 9 on an ISODate "2014-08-11T00:00:00Z".

QUERY:

```
db.restaurants.find( { "grades.1.grade": "A", "grades.1.score": 9, "grades.1.date": ISODate("2014-08-11T00:00:00Z") }, { restaurant_id: 1, name: 1, grades: 1 } );
```

OUTPUT:

Enter a title...

MongoDB

```
1: "A", "grades.1.score": 9, "grades.1.date": ISODate("2014-08-11T00:00:00Z"), { restaurant_id: 1, name: 1, grades: 1 }; Output
```

```
mycompiler_mongodb>
mycompiler_mongodb>

[Execution complete with exit code 0]
```

4.)Write a MongoDB query to find the restaurant Id, name, address and geographical location for those restaurants where 2nd element of coord array contains a value which is more than 42 and upto 52

QUERY:

```
db.restaurants.find({$and : [{"address.coord.1": {$gt : 42}}, {"address.coord.1": {$lte : 52}}]}, {_id:0, restaurant_id:1, name:1, address:1})
```

OUTPUT:

The screenshot shows a MongoDB shell interface. At the top, there is a search bar labeled "Enter a title..." and a dropdown menu set to "MongoDB". Below the search bar are two buttons: a blue one with a gear icon and a red one with a refresh icon. To the right is a green "Run" button with a play icon. The main area is titled "Output" and contains the following text:

```
1{$and : [{"address.coord.1": {$gt : 42}}, {"address.coord.1": {$lte : 52}}]}, {_id:0, restaurant_id:1, name:1, address:1}]] Output
```

```
mycompiler_mongodb>
mycompiler_mongodb>
```

[Execution complete with exit code 0]

5.) Write a MongoDB query to arrange the name of the restaurants in ascending order along with all the columns.

QUERY:

```
db.restaurants.find({}, { _id: 0 }).sort({ name: 1 });
```

OUTPUT:

The screenshot shows a MongoDB shell interface. On the left, there is an input field labeled "Enter a title..." and a toolbar with icons for MongoDB, Help, and a green "Run" button. The code entered is:

```
1 db.restaurants.find({}, { _id: 0 }).sort({ name: 1 });
```

On the right, under the "Output" section, the response is:

```
mycompiler_mongodb>
mycompiler_mongodb>

[Execution complete with exit code 0]
```

6.) Write a MongoDB query to arrange the name of the restaurants in descending along with all the columns.

QUERY:

```
db.restaurants.find({}, { _id: 0 }).sort({ name: 1 })
```

OUTPUT:

The screenshot shows a MongoDB shell interface. On the left, there is an input field labeled "Enter a title..." and a toolbar with icons for MongoDB, Help, and a green "Run" button. The code entered is:

```
1{$and : [{"address.coord.1": {$gt : 42}}, {"address.coord.1": {$lte : 52}}]}, {_id:0, restaurant_id:1, name:1, address:1}]
```

On the right, under the "Output" section, the response is:

```
mycompiler_mongodb>
mycompiler_mongodb>

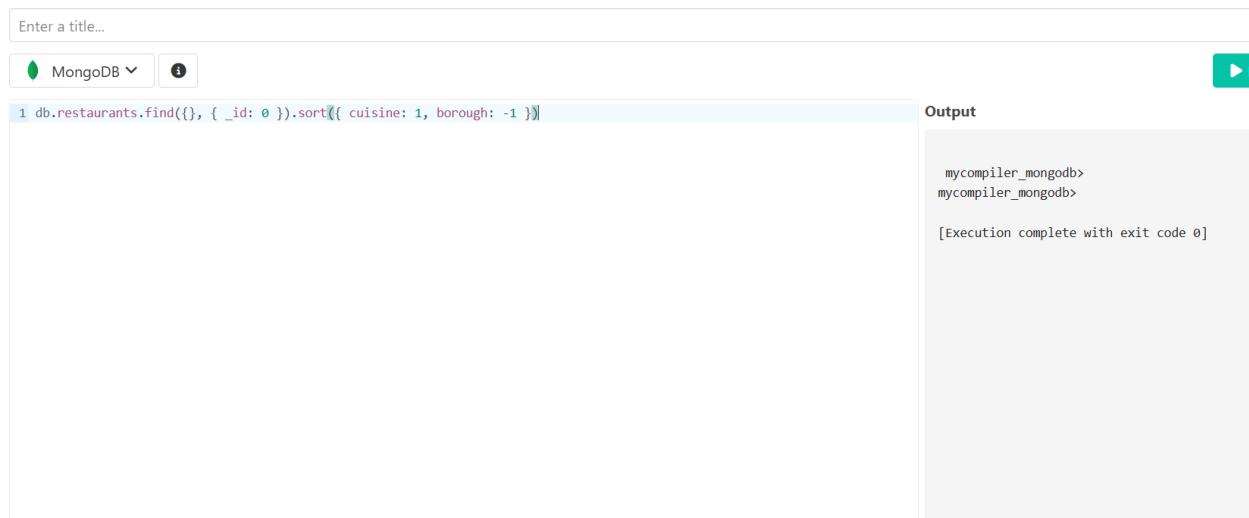
[Execution complete with exit code 0]
```

7.) Write a MongoDB query to arranged the name of the cuisine in ascending order and for that same cuisine borough should be in descending order.

QUERY:

```
db.restaurants.find( {}, { _id: 0 }).sort({ cuisine: 1, borough: -1 })
```

OUTPUT:



The screenshot shows a MongoDB shell interface. On the left, there is an input field with placeholder text "Enter a title...". Below it are two buttons: "MongoDB" with a dropdown arrow and a small info icon. To the right is a "Run" button with a play icon. The main area contains a code block with the query:

```
1 db.restaurants.find( {}, { _id: 0 }).sort({ cuisine: 1, borough: -1 })
```

. To the right of this is an "Output" panel with the text:
mycompiler_mongodb>
mycompiler_mongodb>
[Execution complete with exit code 0]

8.) Write a MongoDB query to know whether all the addresses contains the street or not.

QUERY:

```
db.restaurants.find( { "address.street": { $exists: true, $ne: "" } } )
```

OUTPUT:



The screenshot shows a MongoDB shell interface. On the left, there is an input field with placeholder text "Enter a title...". Below it are two buttons: "MongoDB" with a dropdown arrow and a small info icon. To the right is a "Run" button with a play icon. The main area contains a code block with the query:

```
1 db.restaurants.find( { "address.street": { $exists: true, $ne: "" } } )
```

. To the right of this is an "Output" panel with the text:
mycompiler_mongodb>
mycompiler_mongodb>
[Execution complete with exit code 0]

9.) Write a MongoDB query which will select all documents in the restaurants collection where the coord field value is Double.

QUERY:

```
db.restaurants.find({ "address.coord": { $elemMatch: { $type: "double" } } })
```

OUTPUT:

The screenshot shows a MongoDB shell interface. On the left, there is a search bar labeled 'Enter a title...' and a dropdown menu set to 'MongoDB'. Below the search bar is a toolbar with a green play button icon and a red 'Run' button. The main area contains a code editor with the following content:

```
1 db.restaurants.find({ "address.coord": { $elemMatch: { $type: "double" } } })
```

To the right of the code editor is a panel titled 'Output' containing the results of the query:

```
mycompiler_mongodb>
mycompiler_mongodb>

[Execution complete with exit code 0]
```

10. Write a MongoDB query which will select the restaurant Id, name and grades for those restaurants which returns 0 as a remainder after dividing the score by 7.

QUERY:

```
db.restaurants.find({ "grades.score": { $mod: [7, 0] } }, { restaurant_id: 1, name: 1, grades: 1
});
```

OUTPUT:

Enter a title...

MongoDB Run

```
1 db.restaurants.find({ "grades.score": { $mod: [7, 0] } }, { restaurant_id: 1, name: 1, grades: 1 });
```

Output

```
mycompiler_mongodb>
mycompiler_mongodb>

[Execution complete with exit code 0]
```

11. Write a MongoDB query to find the restaurant name, borough, longitude and attitude and cuisine for those restaurants which contains 'mon' as three letters somewhere in its name.

QUERY:

```
db.restaurants.find({ name: /mon/i }, { name: 1, borough: 1, "address.coord": 1, cuisine: 1 })
```

OUTPUT:

Enter a title...

MongoDB Run

```
1 db.restaurants.find({ name: /mon/i }, { name: 1, borough: 1, "address.coord": 1, cuisine: 1 })
```

Output

```
mycompiler_mongodb>
mycompiler_mongodb>

[Execution complete with exit code 0]
```

12. Write a MongoDB query to find the restaurant name, borough, longitude and latitude and cuisine for those restaurants which contain 'Mad' as first three letters of its name.

QUERY:

```
db.restaurants.find({ name: /^Mad/i }, { name: 1, borough: 1, "address.coord": 1, cuisine: 1 })
```

OUTPUT:

The screenshot shows a MongoDB shell interface. On the left, there is a text input field with placeholder text 'Enter a title...'. Below it are two buttons: 'MongoDB' with a dropdown arrow and a small info icon. To the right is a green 'Run' button with a play icon. The main area contains a code editor with the following content:

```
1 db.restaurants.find({ name: /^Mad/i }, { name: 1, borough: 1, "address.coord": 1, cuisine: 1 })
```

Below the code editor, the number '2' is followed by '3'. To the right, under the heading 'Output', the terminal session shows:

```
mycompiler_mongodb>
mycompiler_mongodb>
```

At the bottom of the output window, the message '[Execution complete with exit code 0]' is displayed.

13. Write a MongoDB query to find the restaurants that have at least one grade with a score of less than 5.

QUERY:

```
db.restaurants.find({ "grades": { $elemMatch: { "score": { $lt: 5 } } } })
```

OUTPUT:

Enter a title...

MongoDB ▾



```
1 db.restaurants.find({ "grades": { $elemMatch: { "score": { $lt: 5 } } } })
```

2

3

Output

```
mycompiler_mongodb>
mycompiler_mongodb>
```

[Execution complete with exit code 0]

14. Write a MongoDB query to find the restaurants that have at least one grade with a score of less than 5 and that are located in the borough of Manhattan.

QUERY:

```
db.restaurants.find({ "grades": { $elemMatch: { "score": { $lt: 5 } } }, "borough": "Manhattan" })
```

OUTPUT:

Enter a title...

MongoDB ▾



```
1 db.restaurants.find({ "grades": { $elemMatch: { "score": { $lt: 5 } } }, "borough": "Manhattan" })
```

2

3

Output

```
mycompiler_mongodb>
mycompiler_mongodb>
```

[Execution complete with exit code 0]

15. Write a MongoDB query to find the restaurants that have at least one grade with a score of less than 5 and that are located in the borough of Manhattan or Brooklyn.

QUERY:

```
db.restaurants.find({ "grades": { $elemMatch: { "score": { $lt: 5 } } }, $or: [{ "borough": "Manhattan" }, { "borough": "Brooklyn" }] })
```

OUTPUT:

The screenshot shows a MongoDB shell interface. At the top, there is a search bar labeled "Enter a title...". Below it, a dropdown menu says "MongoDB" with a "▼" icon, and a small info icon. On the right, there is a green "Run" button with a play icon. The main area contains the following text:

```
1.find({ "grades": { $elemMatch: { "score": { $lt: 5 } } }, $or: [{ "borough": "Manhattan" }, { "borough": "Brooklyn" }] }) Output
2
3
```

On the right side of the interface, there is a terminal window with the following output:

```
mycompiler_mongodb>
mycompiler_mongodb>
[Execution complete with exit code 0]
```

16. Write a MongoDB query to find the restaurants that have at least one grade with a score of less than 5 and that are located in the borough of Manhattan or Brooklyn, and their cuisine is not American.

QUERY:

```
db.restaurants.find({ "grades": { $elemMatch: { "score": { $lt: 5 } } }, $or: [{ "borough": "Manhattan" }, { "borough": "Brooklyn" }], "cuisine": { $ne: "American" } })
```

OUTPUT:

```
Enter a title...
MongoDB ▾ ⓘ Run
1{ "score": { $lt: 5 } }, $or: [{ "borough": "Manhattan" }, { "borough": "Brooklyn" }], "cuisine": { $ne: "American" }}] Output
mycompiler_mongodb>
mycompiler_mongodb>
[Execution complete with exit code 0]
```

17. Write a MongoDB query to find the restaurants that have at least one grade with a score of less than 5 and that are located in the borough of Manhattan or Brooklyn, and their cuisine is not American or Chinese.

QUERY:

```
db.restaurants.find({ "grades": { $elemMatch: { "score": { $lt: 5 } } }, $or: [{ "borough": "Manhattan" }, { "borough": "Brooklyn" }], "cuisine": { $nin: ["American", "Chinese"] } })
```

OUTPUT:

```
Enter a title...
MongoDB ▾ ⓘ Run
1{ $lt: 5 } }, $or: [{ "borough": "Manhattan" }, { "borough": "Brooklyn" }], "cuisine": { $nin: ["American", "Chinese"] })] Output
mycompiler_mongodb>
mycompiler_mongodb>
[Execution complete with exit code 0]
```

18. Write a MongoDB query to find the restaurants that have a grade with a score of 2 and a grade with a score of 6.

QUERY:

```
db.restaurants.find({ $and: [{ "grades.grade": "A", "grades.score": 2 }, { "grades.grade": "A", "grades.score": 6 }] })
```

OUTPUT:



The screenshot shows a MongoDB shell interface. On the left, there is a text input field with placeholder text 'Enter a title...'. Below it are two buttons: 'MongoDB' with a dropdown arrow and a small info icon. To the right is a green 'Run' button with a play icon. The main area contains a code editor with the following content:

```
1 db.restaurants.find({ $and: [{ "grades.grade": "A", "grades.score": 2 }, { "grades.grade": "A", "grades.score": 6 }] })
```

Next to the code editor is a 'Output' tab. The output window shows the command prompt 'mycompiler_mongodb>' followed by '[Execution complete with exit code 0]'. There is also a small note at the bottom right of the output window: 'mycompiler_mongodb> mycompiler_mongodb> [Execution complete with exit code 0]'.

19. Write a MongoDB query to find the restaurants that have a grade with a score of 2 and a grade with a score of 6 and are located in the borough of Manhattan.

QUERY:

```
db.restaurants.find({ $and: [{ "grades.grade": "A", "grades.score": 2 }, { "grades.grade": "A", "grades.score": 6 }], "borough": "Manhattan" })
```

OUTPUT:

Enter a title...

MongoDB ▾



Run

```
1 $and: [{ "grades.grade": "A", "grades.score": 2 }, { "grades.grade": "A", "grades.score": 6 }], "borough": "Manhattan" }]} Output
```

```
mycompiler_mongodb>
```

```
mycompiler_mongodb>
```

```
[Execution complete with exit code 0]
```

20. Write a MongoDB query to find the restaurants that have a grade with a score of 2 and a grade with a score of 6 and are located in the borough of Manhattan or Brooklyn.

QUERY:

```
db.restaurants.find({ $and: [{ "grades.grade": "A", "grades.score": 2 }, { "grades.grade": "A", "grades.score": 6 }], $or: [{ "borough": "Manhattan" }, { "borough": "Brooklyn" }] })
```

OUTPUT:

Enter a title...

MongoDB ▾



Run

```
1.score": 2 }, { "grades.grade": "A", "grades.score": 6 }], $or: [{ "borough": "Manhattan" }, { "borough": "Brooklyn" }] }]} Output
```

```
mycompiler_mongodb>
```

```
mycompiler_mongodb>
```

```
[Execution complete with exit code 0]
```

21. Write a MongoDB query to find the restaurants that have a grade with a score of 2 and a

grade with a score of 6 and are located in the borough of Manhattan or Brooklyn, and their cuisine is not American.

QUERY:

```
db.restaurants.find({ $and: [{ "grades.grade": "A", "grades.score": 2 }, { "grades.grade": "A", "grades.score": 6 }], $or: [{ "borough": "Manhattan" }, { "borough": "Brooklyn" }], "cuisine": { $ne: "American" } })
```

OUTPUT:

The screenshot shows a MongoDB query editor interface. At the top, there is a search bar labeled "Enter a title...". Below it, a dropdown menu is set to "MongoDB" and a "Run" button is visible. The main area displays the MongoDB command and its output. The command is:

```
1 'A", "grades.score": 6 }], $or: [{ "borough": "Manhattan" }, { "borough": "Brooklyn" }], "cuisine": { $ne: "American" } })
```

The output shows the command being run on a database named "mycompiler_mongodb". The response indicates that the command was successful with an exit code of 0.

22. Write a MongoDB query to find the restaurants that have a grade with a score of 2 and a grade with a score of 6 and are located in the borough of Manhattan or Brooklyn, and their cuisine is not American or Chinese.

QUERY:

```
db.restaurants.find({ $and: [{ "grades.grade": "A", "grades.score": 2 }, { "grades.grade": "A", "grades.score": 6 }], $or: [{ "borough": "Manhattan" }, { "borough": "Brooklyn" }], "cuisine": { $nin: ["American", "Chinese"] } })
```

OUTPUT:

The screenshot shows a MongoDB query editor interface. At the top, there is a search bar labeled "Enter a title...". Below it, a dropdown menu is set to "MongoDB" and a "Run" button is visible. The main area displays the MongoDB command and its output. The command is:

```
1 "core": 6 }], $or: [{ "borough": "Manhattan" }, { "borough": "Brooklyn" }], "cuisine": { $nin: ["American", "Chinese"] } })
```

The output shows the command being run on a database named "mycompiler_mongodb". The response indicates that the command was successful with an exit code of 0.

23. Write a MongoDB query to find the restaurants that have a grade with a score of 2 or a grade with a score of 6.

QUERY:

```
db.restaurants.find({ $or: [{ "grades.score": 2 }, { "grades.score": 6 }] })
```

OUTPUT:

The screenshot shows a MongoDB shell interface. On the left, there is a text input field with placeholder text 'Enter a title...'. Below it are two buttons: one with a green icon labeled 'MongoDB' and another with a blue icon. To the right are two buttons: a teal 'Run' button and a blue 'Save' button. The main area contains a code block with the query:

```
1 db.restaurants.find({ $or: [{ "grades.score": 2 }, { "grades.score": 6 }] })
```

. To the right, under the heading 'Output', is a terminal window showing the command prompt 'mycompiler_mongodb>' followed by '[Execution complete with exit code 0]'. The entire interface is set against a dark background.

Evaluation Procedure	Marks awarded
Query(5)	
Execution (5)	
Viva(5)	
Total (15)	
Faculty Signature	

RESULT:

MONGO DB

EX_NO: 20

DATE:

1.) Find all movies with full information from the 'movies' collection that released in the year 1893.

QUERY:

```
db.movies.find({ year: 1893 })
```

OUTPUT:

The screenshot shows a MongoDB shell interface. On the left, there is an input field labeled "Enter a title...". Below it, a dropdown menu is set to "MongoDB" and a help icon is visible. On the right, a "Run" button is shown with a play icon. The code input area contains the command: `1 db.movies.find({ year: 1893 })`. To the right, under the heading "Output", the response is displayed as follows:

```
mycompiler_mongodb>
mycompiler_mongodb>
[Execution complete with exit code 0]
```

2.) Find all movies with full information from the 'movies' collection that have a runtime greater than 120 minutes.

QUERY:

```
db.movies.find({ runtime: { $gt: 120 } })
```

OUTPUT:

The screenshot shows a MongoDB shell interface. On the left, there is an input field labeled "Enter a title...". Below it, a dropdown menu is set to "MongoDB" and a help icon is visible. On the right, a "Run" button is shown with a play icon. The code input area contains the command: `1 db.movies.find({ runtime: { $gt: 120 } })`. To the right, under the heading "Output", the response is displayed as follows:

```
mycompiler_mongodb>
mycompiler_mongodb>
[Execution complete with exit code 0]
```

3.) Find all movies with full information from the 'movies' collection that have "Short" genre.

QUERY:

```
db.movies.find({ genres: 'Short' })
```

OUTPUT:

The screenshot shows a MongoDB shell interface. On the left, there is a text input field with placeholder text "Enter a title...". Below it are two buttons: "MongoDB" with a dropdown arrow and a small info icon. To the right is a green "Run" button with a play icon and the text "Ctrl+Enter". The main area contains a code snippet: "1 db.movies.find({ genres: 'short' })". To the right, under the heading "Output", the results are displayed: "mycompiler_mongodb>" followed by a blank line, another "mycompiler_mongodb>" line, and "[Execution complete with exit code 0]" at the bottom.

4.) Retrieve all movies from the 'movies' collection that were directed by "William K.L. Dickson" and include complete information for each movie.**QUERY:**

```
db.movies.find({ directors: 'William K.L. Dickson' })
```

OUTPUT:

The screenshot shows a MongoDB shell interface. On the left, there is a text input field with placeholder text "Enter a title...". Below it are two buttons: "MongoDB" with a dropdown arrow and a small info icon. To the right is a green "Run" button with a play icon and the text "Ctrl+Enter". The main area contains a code snippet: "1 db.movies.find({ directors: 'William K.L. Dickson' })". To the right, under the heading "Output", the results are displayed: "mycompiler_mongodb>" followed by a blank line, another "mycompiler_mongodb>" line, and "[Execution complete with exit code 0]" at the bottom.

5.) Retrieve all movies from the 'movies' collection that were released in the USA and include complete information for each movie.**QUERY:**

```
db.movies.find({ countries: 'USA' })
```

OUTPUT:

The screenshot shows a MongoDB shell interface. On the left, there is a text input field with placeholder text "Enter a title...". Below it are two buttons: one with a green leaf icon labeled "MongoDB" and another with an info icon. To the right is a "Run" button with a play icon. The main area contains the command: `1 db.movies.find({ countries: 'USA' })`. On the far right, under the heading "Output", the response is shown: `mycompiler_mongodb>`, `mycompiler_mongodb>`, and [Execution complete with exit code 0].

6.) Retrieve all movies from the 'movies' collection that have complete information and are rated as "UNRATED".

QUERY:

```
db.movies.find({ rated: 'UNRATED' })
```

OUTPUT:

The screenshot shows a MongoDB shell interface. On the left, there is a text input field with placeholder text "Enter a title...". Below it are two buttons: one with a green leaf icon labeled "MongoDB" and another with an info icon. To the right is a "Run" button with a play icon. The main area contains the command: `1 db.movies.find({ rated: 'UNRATED' })`. On the far right, under the heading "Output", the response is shown: `mycompiler_mongodb>`, `mycompiler_mongodb>`, and [Execution complete with exit code 0].

7.) Retrieve all movies from the 'movies' collection that have complete information and have received more than 1000 votes on IMDb.

QUERY:

```
db.movies.find({ 'imdb.votes': { $gt: 1000 } })
```

OUTPUT:

Enter a title...

MongoDB ▾

Ctrl+Enter

Run

```
1 db.movies.find({ 'imdb.votes': { $gt: 1000 } })
```

Output

```
mycompiler_mongodb>
mycompiler_mongodb>

[Execution complete with exit code 0]
```

8.) Retrieve all movies from the 'movies' collection that have complete information and have an IMDb rating higher than 7.

QUERY:

```
db.movies.find({ 'imdb.rating': { $gt: 7 } })
```

OUTPUT:

Enter a title...

MongoDB ▾

Run

```
1 db.movies.find({ 'imdb.rating': { $gt: 7 } })
```

Output

```
mycompiler_mongodb>
mycompiler_mongodb>

[Execution complete with exit code 0]
```

9.) Retrieve all movies from the 'movies' collection that have complete information and have a viewer rating higher than 4 on Tomatoes.

QUERY:

```
db.movies.find({ 'tomatoes.viewer.rating': { $gt: 4 } })
```

OUTPUT:

The screenshot shows a MongoDB shell interface. On the left, there is a search bar labeled "Enter a title...". Below it are two buttons: "MongoDB" with a dropdown arrow and a small info icon. To the right is a green "Run" button with a play icon. In the main area, a command is entered: `db.movies.find({ 'tomatoes.viewer.rating': { $gt: 4 } })`. The output window on the right shows the results of the query:

```
mycompiler_mongodb>
mycompiler_mongodb>
[Execution complete with exit code 0]
```

10.) Retrieve all movies from the 'movies' collection that have received an award.

QUERY:

```
db.movies.find({ 'awards.wins': { $gt: 0 } })
```

OUTPUT:

The screenshot shows a MongoDB shell interface. On the left, there is a search bar labeled "Enter a title...". Below it are two buttons: "MongoDB" with a dropdown arrow and a small info icon. To the right is a green "Run" button with a play icon. In the main area, a command is entered: `db.movies.find({ 'awards.wins': { $gt: 0 } })`. The output window on the right shows the results of the query:

```
mycompiler_mongodb>
mycompiler_mongodb>
[Execution complete with exit code 0]
```

11.) Find all movies with title, languages, released, directors, writers, awards, year, genres, runtime, cast, countries from the 'movies' collection in MongoDB that have at

least one nomination.

QUERY:

```
db.movies.find( { 'awards.nominations': { $gt: 0 } }, { title: 1, languages: 1, released: 1, directors: 1, writers: 1, awards: 1, year: 1, genres: 1, runtime: 1, cast: 1, countries: 1 })
```

OUTPUT:

The screenshot shows a MongoDB shell interface. On the left, there is a search bar labeled "Enter a title..." and a dropdown menu set to "MongoDB". Below the search bar are two buttons: a green one with a gear icon and a blue one with a question mark icon. To the right of these buttons is a teal "Run" button. The main area contains a command line and its output. The command is:

```
1, languages: 1, released: 1, directors: 1, writers: 1, awards: 1, year: 1, genres: 1, runtime: 1, cast: 1, countries: 1 })
```

The output shows the results of the query:

```
mycompiler_mongodb>
mycompiler_mongodb>
[Execution complete with exit code 0]
```

At the bottom right of the interface, there is a small red button with a white arrow pointing right, and the text "Explore design possibilities with Figma" followed by "easily translate your work into code".

12.) Find all movies with title, languages, released, directors, writers, awards, year, genres, runtime, cast, countries from the 'movies' collection in MongoDB with cast including "Charles Kayser".

QUERY:

```
db.movies.find( { cast: 'Charles Kayser' }, { title: 1, languages: 1, released: 1, directors: 1, writers: 1, awards: 1, year: 1, genres: 1, runtime: 1, cast: 1, countries: 1 })
```

OUTPUT:

```
Enter a title...  
MongoDB Run  
1, languages: 1, released: 1, directors: 1, writers: 1, awards: 1, year: 1, genres: 1, runtime: 1, cast: 1, countries: 1 })|| Output  
mycompiler_mongodb>  
mycompiler_mongodb>  
[Execution complete with exit code 0]
```

13.) Retrieve all movies with title, languages, released, directors, writers, countries from the 'movies' collection in MongoDB that released on May 9, 1893.

QUERY:

```
db.movies.find( { released: ISODate("1893-05-09T00:00:00.000Z") }, { title: 1, languages: 1, released: 1, directors: 1, writers: 1, countries: 1 } )
```

OUTPUT:

```
Enter a title... Ctrl+Enter  
MongoDB Run  
1: ISODate("1893-05-09T00:00:00.000Z" ), { title: 1, languages: 1, released: 1, directors: 1, writers: 1, countries: 1 })|| Output  
mycompiler_mongodb>  
mycompiler_mongodb>  
[Execution complete with exit code 0]
```

14.) Retrieve all movies with title, languages, released, directors, writers, countries from the 'movies' collection in MongoDB that have a word "scene" in the title.

QUERY:

```
db.movies.find( { title: /scene/i }, { title: 1, languages: 1, released: 1, directors: 1, writers: 1, countries: 1 } )
```

OUTPUT:



The screenshot shows a MongoDB shell interface. On the left, there is a text input field with placeholder text "Enter a title...". Below it are two buttons: "MongoDB" with a dropdown arrow and a small info icon. To the right is a "Run" button with a play icon. The main area contains a code block with the query: "1 db.movies.find({ title: /scene/i }, { title: 1, languages: 1, released: 1, directors: 1, writers: 1, countries: 1 })". To the right of the code is a "Output" section. The output shows the command "mycompiler_mongodb>" followed by "mycompiler_mongodb>". At the bottom of the output section, a message reads "[Execution complete with exit code 0]".

Evaluation Procedure	Marks awarded
Query(5)	
Execution (5)	
Viva(5)	
Total (15)	
Faculty Signature	

RESULT: